

Mobicents JAIN SLEE Management Console User Guide

by Alexandre Mendonça

Preface	v
1. Document Conventions	v
1.1. Typographic Conventions	v
1.2. Pull-quote Conventions	vii
1.3. Notes and Warnings	vii
2. Provide feedback to the authors!	viii
1. Introduction to Mobicents JAIN SLEE Management Console	1
2. Installing Mobicents JAIN SLEE Management Console	3
2.1. Pre-Install Requirements and Prerequisites	3
2.1.1. Hardware Requirements	3
2.1.2. Software Prerequisites	3
2.2. Install Mobicents JAIN SLEE Management Console	3
2.3. Uninstalling Mobicents JAIN SLEE Management Console	3
2.4. Building Mobicents JAIN SLEE Management Console from source	4
3. Configuring and Running Mobicents JAIN SLEE Management Console	7
3.1. Running Mobicents JAIN SLEE Management Console	7
3.2. Configuring Mobicents JAIN SLEE Management Console	7
4. Managing Mobicents JAIN SLEE with Mobicents JAIN SLEE Management Console... ..	9
4.1. "SLEE" View	9
4.2. "Deployable Units" View	11
4.2.1. View Deployable Units Details	11
4.2.2. Search Deployable Units	13
4.2.3. Install Deployable Units	13
4.2.4. Uninstall Deployable Units	14
4.3. "Components" View	15
4.3.1. View Components Details	15
4.3.2. Search Components	18
4.4. "Services" View	20
4.4.1. View Service Details	20
4.4.2. View and Change Service State	21
4.4.3. Service Usage Parameters	22
4.5. "Resources" View	23
4.5.1. View Resource Adaptor Details	24
4.5.2. View and Change Resource Adaptor Entities State	25
4.5.3. Create a Resource Adaptor Entity	25
4.5.4. Remove a Resource Adaptor Entity	26
4.5.5. View Resource Adaptor Entity Details	26
4.5.6. View and Change Resource Adaptor Entity Configuration Properties	26
4.5.7. View and Change Resource Adaptor Entity Links	27
4.5.8. Resource Adaptor Usage Parameters	27
5. Monitoring Mobicents JAIN SLEE with Mobicents JAIN SLEE Management Console. 31	
5.1. "Activities" View	31
5.1.1. View Activities Details	32
5.1.2. Query Activity Context Liveness	34

5.2. "Alarms" View	35
5.2.1. View Alarms Details	36
5.2.2. Clear an Alarm	37
A. Java Development Kit (JDK): Installing, Configuring and Running	39
B. Setting the JBOSS_HOME Environment Variable	43
C. Revision History	47
Index	49

Preface

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/) [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 Of 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 Mobicents JAIN SLEE Management Console

Mobicents JAIN SLEE Management Console is an application which lets an administrator user interact with the SLEE in order to perform management functions.

The console is a J2EE compliant web-based application deployed in the running instance of Apache Tomcat inside JBoss AS. An administrator can connect to this console via HTTP through a common web browser.

Mobicents JAIN SLEE Management Console allows the administrator to perform most common management operations, such as deployment and installation of JAIN SLEE Deployable Units, JAIN SLEE Component Management and Configuration, etc. as well as some monitoring task such as observing and inspecting JAIN SLEE Activities.

Installing Mobicents JAIN SLEE Management Console

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 Mobicents JAIN SLEE 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 Mobicents JAIN SLEE Management Console platform will run on the same hardware that the JBoss Application Server runs on.

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 JAIN SLEE Management Console. 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](#).

AJAX/CSS3 Compatible Browser

Mobicents JAIN SLEE Management Console client side components use AJAX and CSS3, therefore the use of an up-to-date browser is recommended.

2.2. Install Mobicents JAIN SLEE Management Console

The Mobicents JAIN SLEE Management Console is pre-installed in Mobicents JAIN SLEE binary release. The file `slee-management-console.war` is present in `JBOSS_HOME/server/<node>/deploy/` directory.

2.3. Uninstalling Mobicents JAIN SLEE Management Console

To uninstall Mobicents JAIN SLEE Management Console simply delete the `slee-management-console.war` file from `JBOSS_HOME/server/<node>/deploy/` directory.

2.4. Building Mobicents JAIN SLEE Management Console from source

Mobicents JAIN SLEE Management Console can be built from source by using Maven build tool, following these steps:

1. Downloading the source code



Important

Subversion is used to manage its source code. Instructions for using Subversion, including install, can be found at <http://svnbook.red-bean.com>

Use SVN to checkout a specific release source, the base URL is , then add the specific release version, eg: 2.0.0.BETA1.

```
[usr]$ svn co http://mobicents.googlecode.com/svn/tags/servers/jain-slee/2.x.y/tools/slee-  
management-console/2.0.0.BETA1/ slee-management-console-2.0.0.BETA1
```

If desired, it could be downloaded from the SVN trunk, but this is not always stable or even guaranteed to build correctly. To do so, use:

```
[usr]$ svn co http://mobicents.googlecode.com/svn/trunk/servers/jain-slee/tools/slee-  
management-console/ slee-management-console
```

2. Building the source code



Important

Maven 2.0.9 (or higher) is used to build the release. Instructions for using Maven2, including install, can be found at <http://maven.apache.org>



Important

Make sure to have the JBOSS_HOME environment variable set and pointing to the desired Mobicents installation.

Use Maven to build the binaries.

```
[usr]$ cd slee-management-console-2.0.0.BETA1  
[usr]$ mvn clean install
```

Once the process finishes you should have the `slee-management-console.war` file in the `JBOSS_HOME/server/<node>/deploy/` directory.

Configuring and Running Mobicents JAIN SLEE Management Console

3.1. Running Mobicents JAIN SLEE Management Console

Mobicents JAIN SLEE Management Console starts and stops with Mobicents JAIN SLEE. It can be accessed via web browser at the URL: `http://IP_ADDRESS:8080/slee-management-console/`.

3.2. Configuring Mobicents JAIN SLEE Management Console

Mobicents JAIN SLEE Management Console has no configuration options.

Managing Mobicents JAIN SLEE with Mobicents JAIN SLEE Management Console

4.1. "SLEE" View

In this view, it is possible to view the ID of the server, such as name, vendor, version and codename and also the SLEE state.

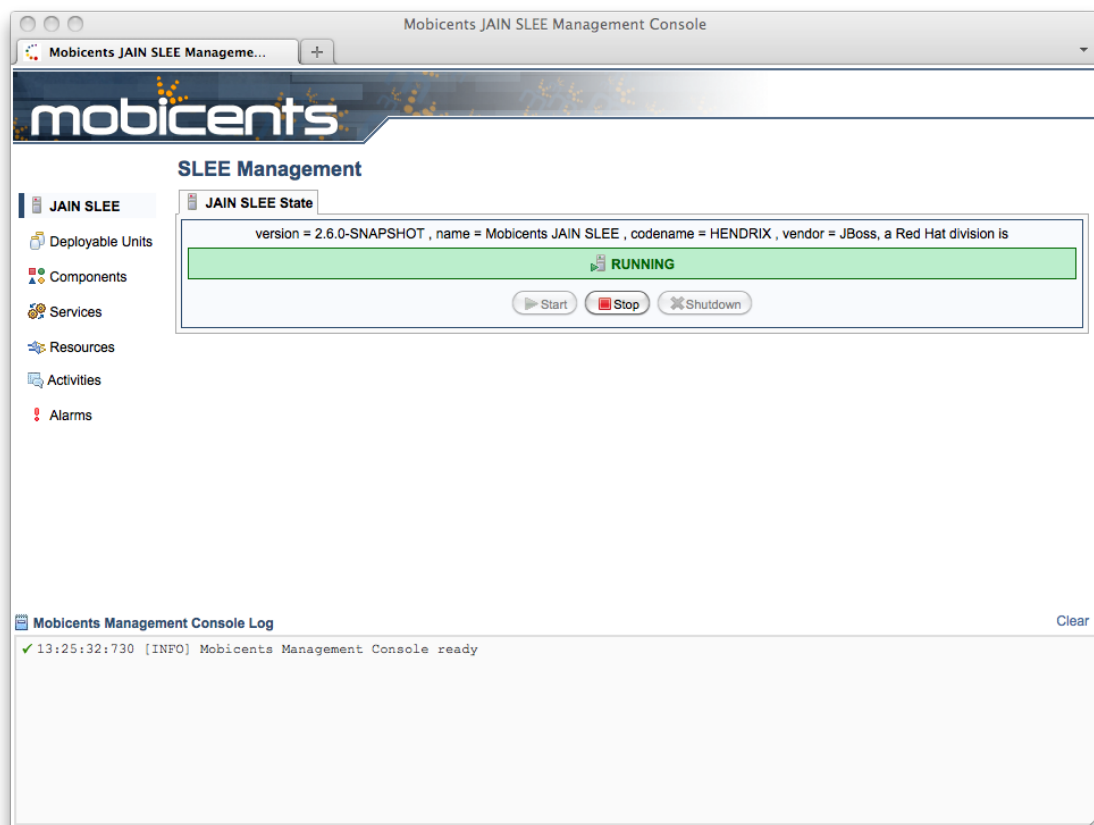


Figure 4.1. SLEE Main View

The following states can be observed:

- STOPPED

The SLEE environment is configured and initialized, ready to be started. This means resource adaptor objects for resource adaptor entities in the `ACTIVE` state are loaded and initialized, and

SBBs corresponding to Services in the `ACTIVE` state are loaded and ready to be instantiated. However the entire event-driven subsystem is idle. Resource adaptor entities and the SLEE are not actively producing events, and the event router is not operating. SBB entities are not created in this state.

- **STARTING**

Any vendor-specific starting state tasks may be performed here. Services in the `ACTIVE` state are made ready to receive events, but SBB entities are still not created in this state. The SLEE spontaneously moves out of this state when (a) startup tasks are complete, which causes transition to the `RUNNING` state; (b) some startup task fails, which causes transition to the `STOPPING` state.

- **RUNNING**

Resource adaptor objects that are in the `ACTIVE` (or `STOPPING`) state are actively firing events as conditions dictate. The SLEE may also be firing events. The event router is instantiating SBB entities and delivering events to them as required.

- **STOPPING**

This state is identical to the `RUNNING` state except no new Activity objects are accepted by the SLEE from resource adaptor objects, and no new Activity objects are created by the SLEE. If this state is reached from the `STARTING` state, there will be no Activity objects and transition to the `STOPPED` state should immediately occur. If this state is reached from the `RUNNING` state, any existing Activity objects are allowed to end (subject to an arbitrary vendor-specified timeout). Once all Activity objects generated by a particular resource adaptor entity have ended, resource adaptor objects of that resource adaptor entity transition to the `INACTIVE` state. The SLEE transitions out of the `STOPPING` state once all Activity objects have ended and all resource adaptor objects have transitioned to the `INACTIVE` state.

It is also possible to manage the SLEE state from this view, by using the following buttons:

- **Start**

In order to use this button the SLEE must be in the `STOPPED` state. It will move the SLEE to the `STARTING` state and once the start operations complete, switch to the `RUNNING` state.

- **Stop**

In order to use this button the SLEE must be in the `STARTED` state. It will move the SLEE to the `STOPPING` state and once the stop operations complete, switch to the `STOPPED` state.

- **Shutdown**

In order to use this button the SLEE must be in the `STOPPED` state. It will shutdown completely the SLEE container as well as the JBoss Application Server.

4.2. "Deployable Units" View

In this view, it is possible to view, search, install and uninstall Deployable Units in the SLEE. A list showing the deployable units, identified by their filename, is shown.

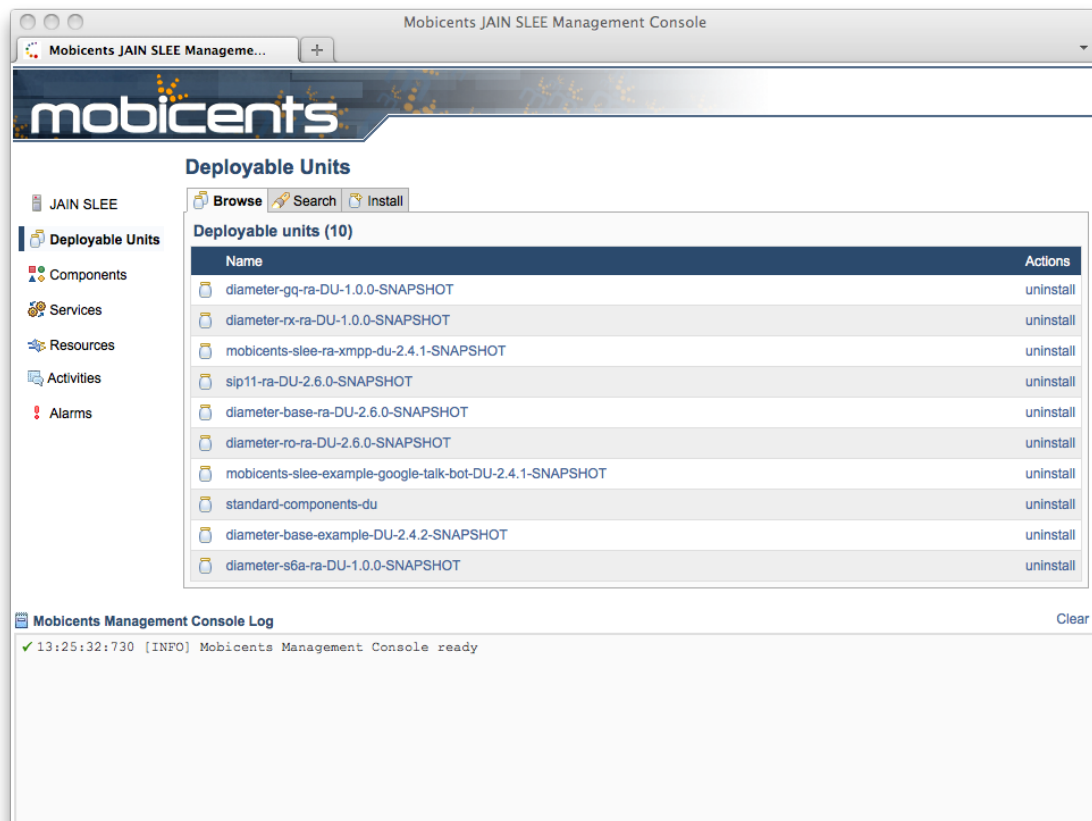


Figure 4.2. Deployable Units Main View

4.2.1. View Deployable Units Details

It is possible to see detailed information about the deployable units by clicking on it's name.

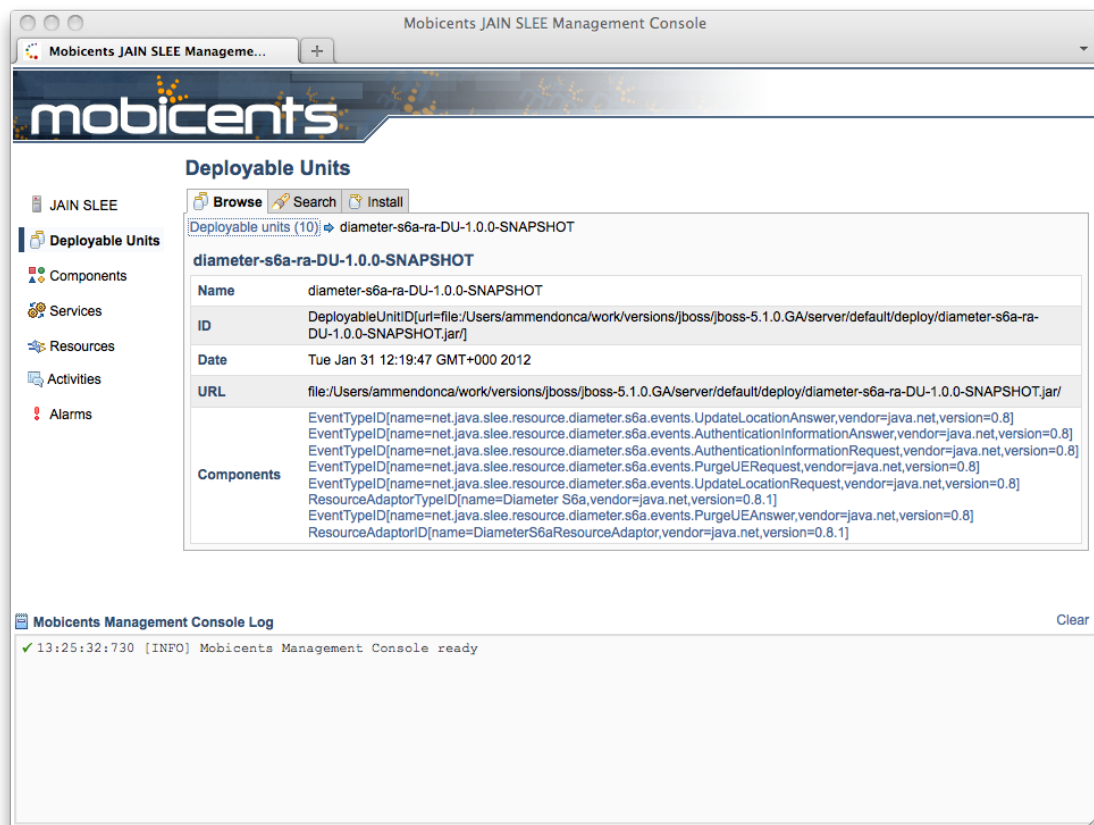


Figure 4.3. Deployable Units Details

The following fields are shown:

- Name

The Deployable Unit name, extracted from it's filename.

- ID

The SLEE identifier for this Deployable Unit.

- Date

The deployment date for this Deployable Unit.

- URL

The URL from where this deployable unit was deployed from.

- Components

The SLEE components included and deployed with this deployable unit.

4.2.2. Search Deployable Units

In order to find specific deployable units, use the **Search** tab, where it is possible to specify a string to search for and click the **Search** button to list the DUs for which the name matches the given string.

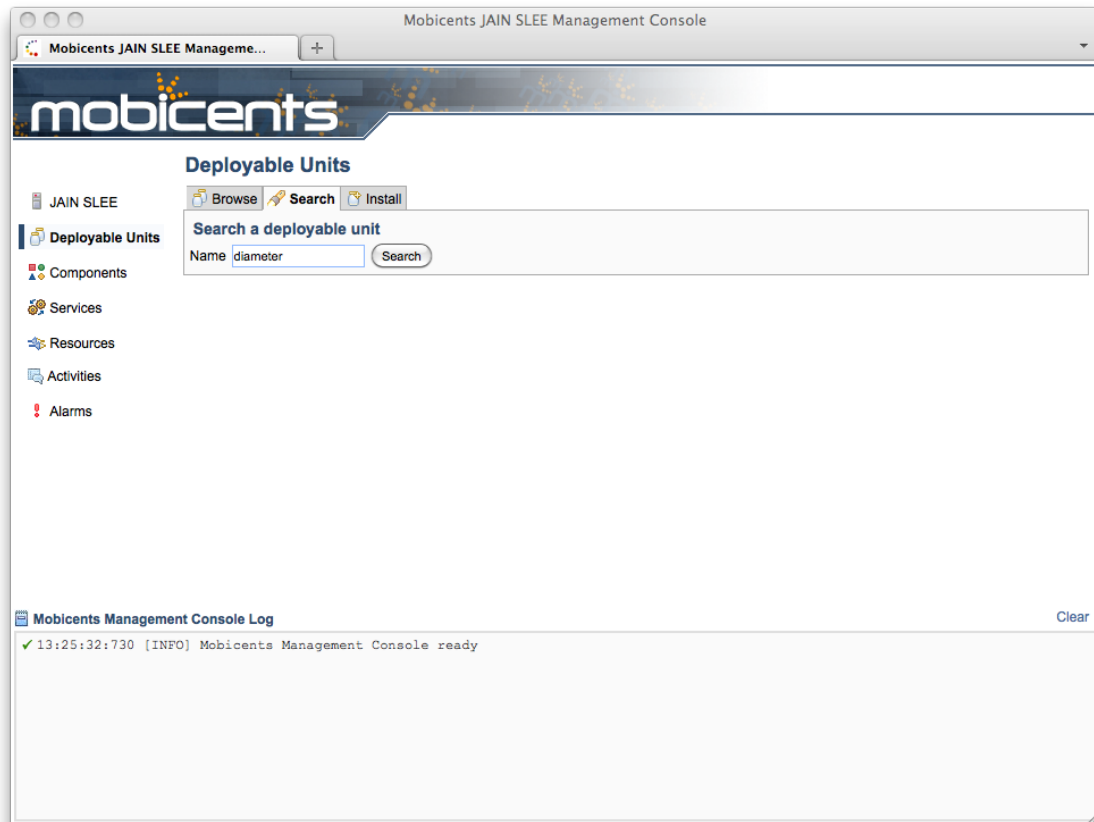


Figure 4.4. Deployable Units Search View



Regular Expression not supported in Deployable Unit Search

When searching for deployable units, it is not possible to use regular expressions. So using "diam*" may not return anything, while "diameter" may return something.

4.2.3. Install Deployable Units

A deployable unit can be installed to SLEE by going to the **Install** tab, clicking the **Choose File** button, browsing to the desired deployable unit file and clicking the **Install** button.

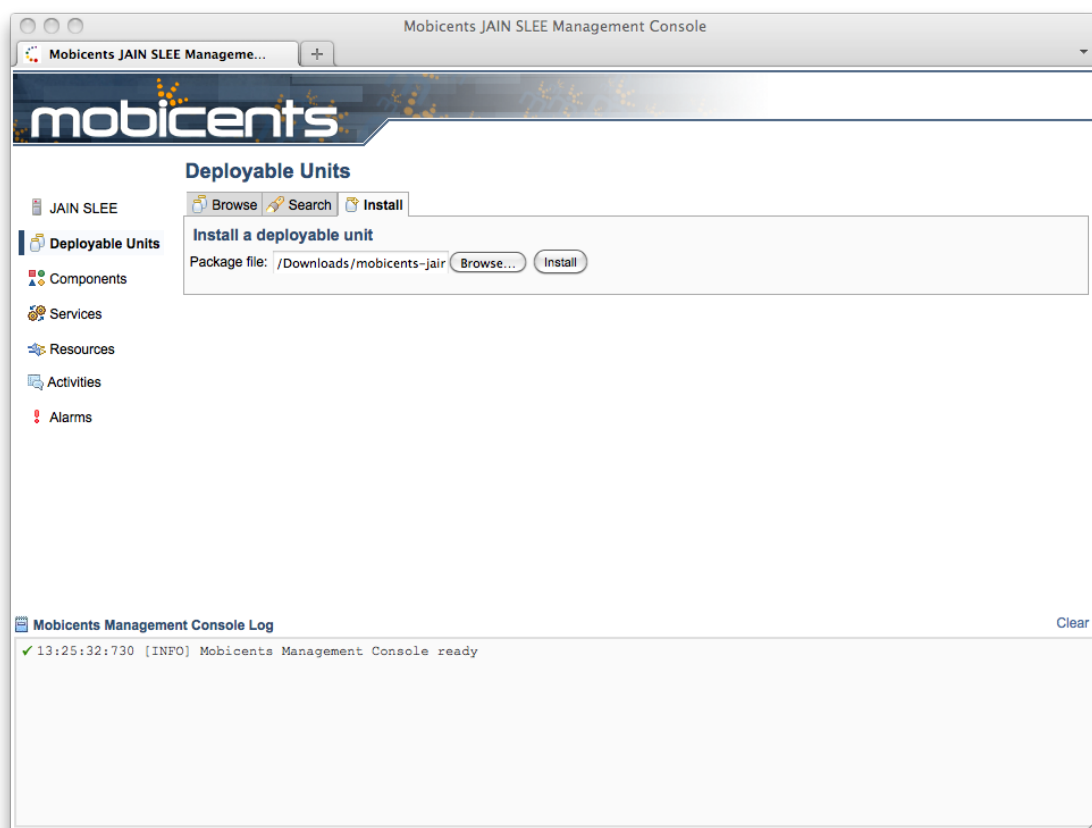


Figure 4.5. Deployable Units Install View



Deployable Unit Installation

Please note that only the deployable unit will be installed, no further operations will be performed. If you need to, you will need to activate services, create Resource Adaptor Entities and Links, etc.

4.2.4. Uninstall Deployable Units

A deployable unit can be uninstalled from SLEE in the list view (both in **Browse** or **Search** tabs) by clicking the **uninstall** link.



Deployable Unit Uninstall

Please note that the deployable unit will be uninstalled, only if there are no components depending on it, no Links or Entities in case of Resource Adaptors and in case of services, they must be deactivated.

4.3. "Components" View

In this view, the components installed in SLEE are shown, grouped by their types. It is possible to view the components, their details and search for specific components.

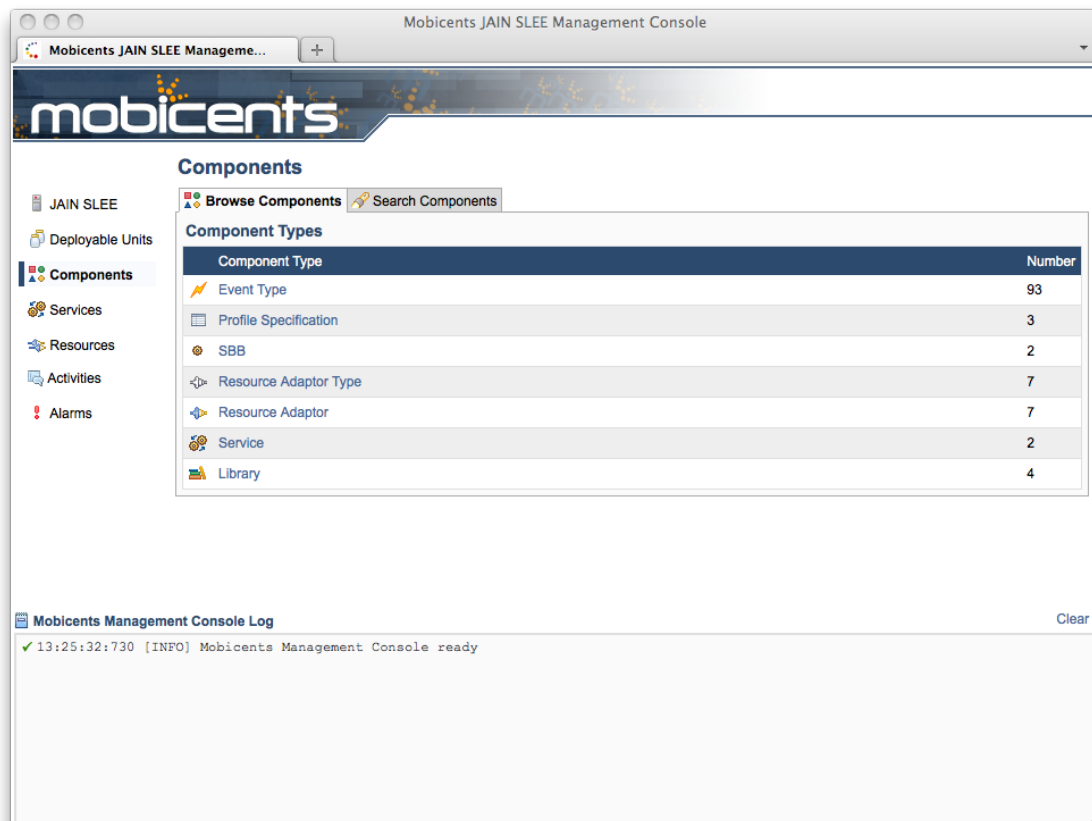


Figure 4.6. Components Main View

4.3.1. View Components Details

In the **Browse Components** tab a list of components is shown, grouped by their type. When clicking on one of the Component Types, a list of the available components of that type is shown, identified by their Name, Vendor and Version.

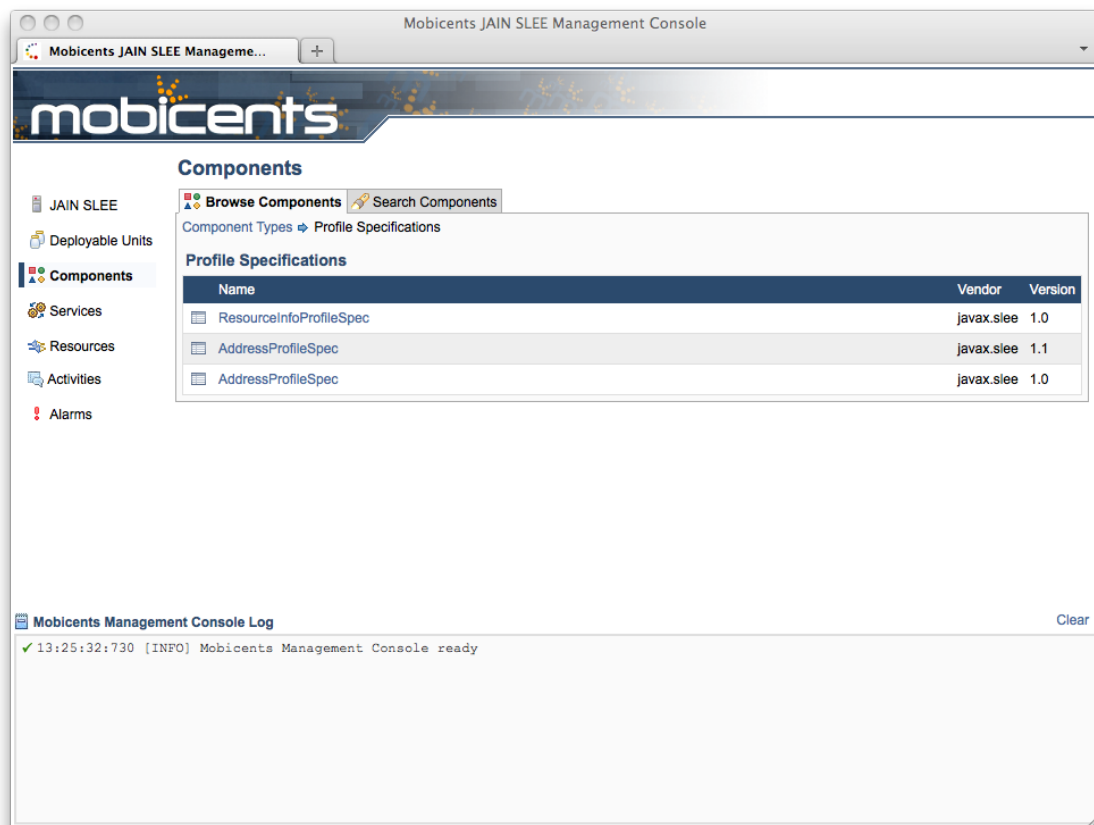


Figure 4.7. Selected Component Type View

If one of this components is clicked, it's details are shown. The following component types and details are available:

The following details are common to all component types:

- Name

The SLEE component Name identifier.

- ID

The SLEE component identifier, built from the component type, name, vendor and version.

- Vendor

The SLEE component Vendor identifier.

- Version

The SLEE component Version identifier.

- Source

The path to the jar containing this component, inside the deployable unit jar.

- Deployable Unit

The deployable unit used to deploy this component.

- Library References

The libraries this component refers, if any.

The following are specific to each component type:

- Event Type

- Event Class Name

The class name for this event type.

- Profile Specification

- CMP Interface Name

The interface name for the Profile CMP.

- SBB

- Address Profile Specification

The profile specification to be used as the Address Profile Specification for this SBB.

- Event Types

The event types referenced by this SBB, either for Fire and/or Receive.

- Profile Specifications

The profile specs referenced by this SBB.

- Resource Adaptor Entity Links

The resource adaptor links required by this SBB.

- Resource Adaptor Types

The resource adaptor types referenced by this SBB, either for Fire and/or Receive.

- SBBs

The child SBBs this SBB has.

- Resource Adaptor Type

- Event Types

The event types referenced by this Resource Adaptor Type, that it can fire.

- Resource Adaptor

- Event Types

The Resource Adaptor Type implemented by this Resource Adaptor.

- Service

- Address Profile Table

The Address Profile Table name to be used by this service, if any.

- Resource Info Profile Table

The Resource Info Profile Table name to be used by this service, if any.

- Root SBB

The SBB to be used as the Root SBB for this service.

- Library

- Library Jars

The JARs exposed by this Library component.

4.3.2. Search Components

In order to find specific components, use the **Search** tab, where it is possible to specify a string to search for and click the **Search** button to list the components for which the name matches the given string.

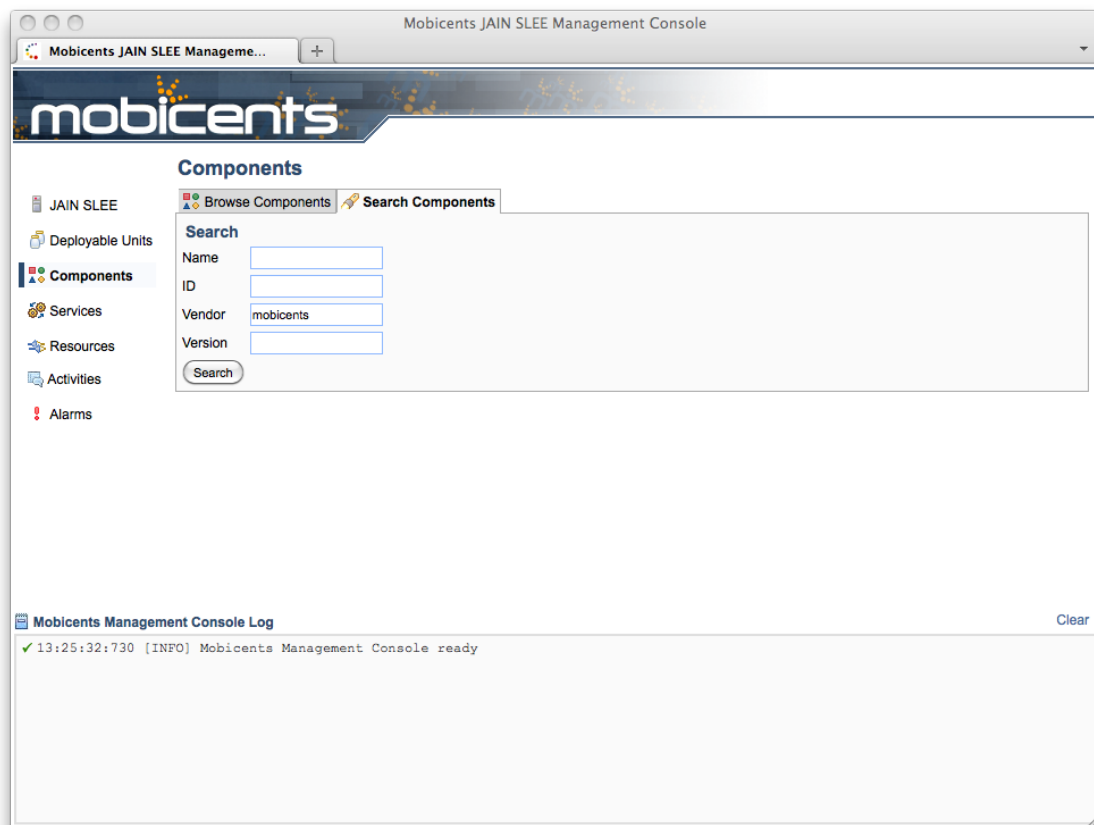


Figure 4.8. Components Search View



Regular Expression not supported in Component Search

When searching for components, it is not possible to use regular expressions. So using "diam*" may not return anything, while "diameter" may return something.

4.4. "Services" View

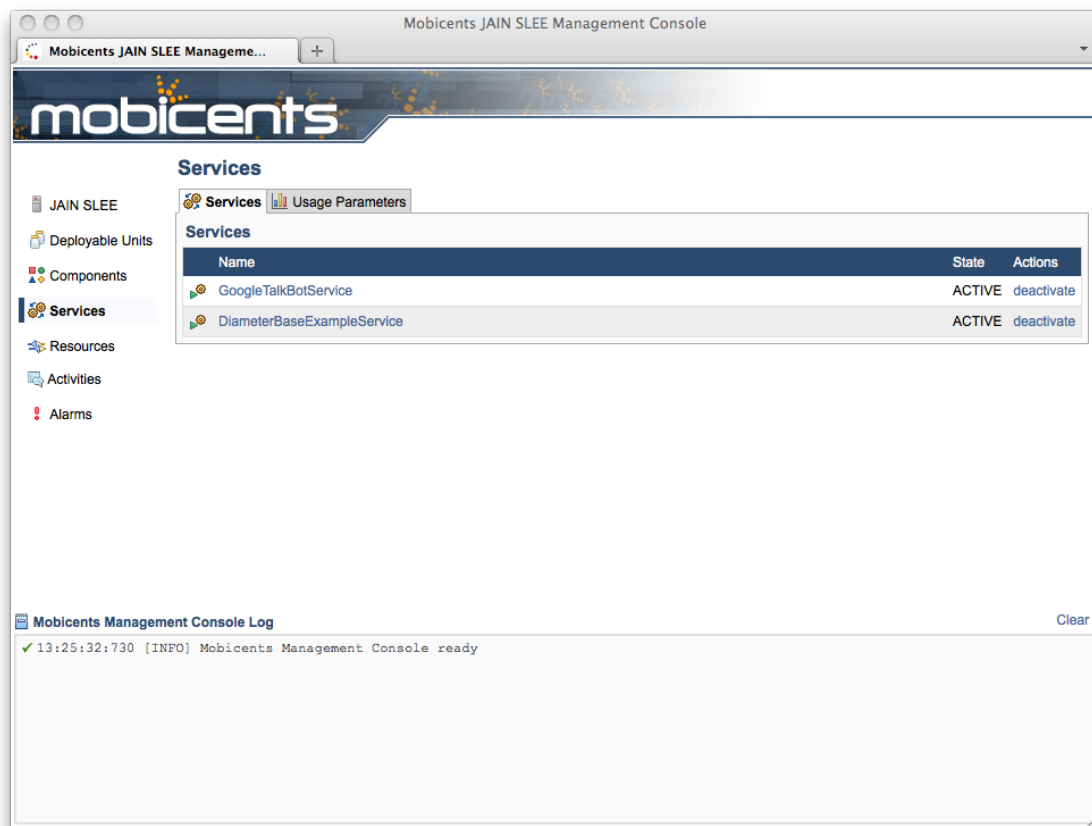


Figure 4.9. Services Main View

In this view, the services present in SLEE are shown, along with their state. It is possible to view the services, their details, view and change the service state.

4.4.1. View Service Details

In the **Services** tab the list of services is shown. When clicking on one of the services' name, it's details are shown. The following details are displayed:

- Name
The SLEE component Name identifier.
- ID
The SLEE component identifier, built from the component type, name, vendor and version.
- Vendor
The SLEE component Vendor identifier.
- Version

The SLEE component Version identifier.

- Source

The path to the jar containing this component, inside the deployable unit jar.

- Deployable Unit

The deployable unit used to deploy this component.

- Library References

The libraries this component refers, if any.

- Address Profile Table

The Address Profile Table name to be used by this service, if any.

- Resource Info Profile Table

The Resource Info Profile Table name to be used by this service, if any.

- Root SBB

The SBB to be used as the Root SBB for this service. This entry is clickable and will lead to the details view for this SBB component.

4.4.2. View and Change Service State

When the existing services list is shown, the service state is presented, it can be:

- INACTIVE

The Service has been installed successfully and is ready to be activated. The Service is not running, i.e. root SBB entities of the Service's root SBB will not be created to process events.

In this state it is possible to activate the service by clicking the **activate** link.

- ACTIVE

The Service has been activated, i.e. it is running. The SLEE will create root SBB entities of the Service's root SBB to receive initial events and invoke SBB entities in the SBB entity trees of the Service.

In this state it is possible to deactivate the service by clicking the **deactivate** link.

- STOPPING

The Service is being deactivated. However, some SBB entity trees of the Service still exist in the SLEE and have not completed their processing. The SLEE is waiting for the SBB entities in these SBB entity trees to complete processing so that they can be reclaimed. An SBB entity

has completed processing and can be reclaimed when it and all of its child SBB entities are no longer attached to any Activity Context.

No state changing operations are available in this state, as this is a transition state, from ACTIVE to INACTIVE.

4.4.3. Service Usage Parameters

In the **Usage Parameters** tab is possible to view and manage the service SBBs Usage Parameters. In this screen, select a Service and an SBB and if a SBB Usage Parameter Interface is defined, this view will present the default and other available Usage Parameter Sets.

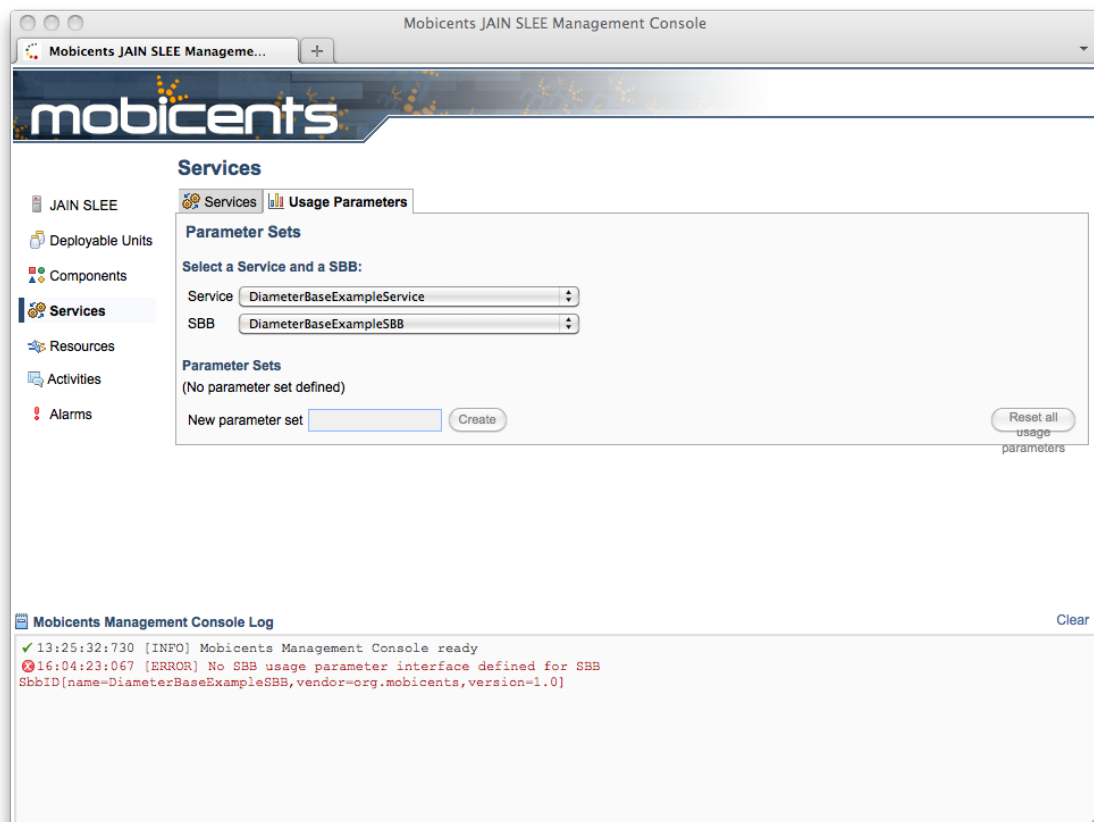


Figure 4.10. Services Usage Parameters View

4.4.3.1. View Service Usage Parameters

In order to view the Usage Parameters for a given set click it's name. The following types are displayed:

- Counter Type

A counter-type usage parameter can be incremented or decremented. The current (approximate) value of the counter-type usage parameter is displayed under the **Value** column is shown.

- Sample Type

A sample-type usage parameter accumulates sample data, when a SLEE Component submit samples to the usage parameter. The current (approximate) minimum, maximum, mean, and the number of sample values added to the sample-type usage parameter are shown.



No automatic refresh

In order to keep a low impact on the SLEE no real-time update is performed with regards to the Usage Parameters. To update the displayed values, please click the **refresh** link.

4.4.3.2. Reset Service Usage Parameters

It is possible to reset a single Usage Parameter, when viewing the Service Usage Parameters list, by clicking the **reset** link, next to each, both for Counter and Sample type usage parameters.

In order to reset all the Usage Parameters for a given service, in the main Usage Parameters view click the **Reset all usage parameters** button and all will be reset.

4.4.3.3. Create Named Service Usage Parameters Set

In addition to the default usage parameter set created by the SLEE, it is possible to create additional usage parameter sets for the same notification source, which must have a name. To create one, in the main Usage Parameters view, fill the **New parameter set** with the name and press the **Create** button and the new set is created.

4.4.3.4. Remove Named Service Usage Parameters Set

To remove a created Usage Parameter Set, in the main Usage Parameters view, under the **Parameter Sets** list, in the **Actions** column, click the **remove** link and the set will be removed.



The "default" Usage Parameters Set is not removable

Only named Usage Parameters Sets can be removed, the "default" will always exist when there's a Usage Parameter Interface defined for this component.

4.5. "Resources" View

In this view, the resource adaptors present in SLEE are shown, identified by their name, vendor, version and RA Type implemented. It is possible to view the resources, their details, view and manage Resource Adaptor Entities and Links.

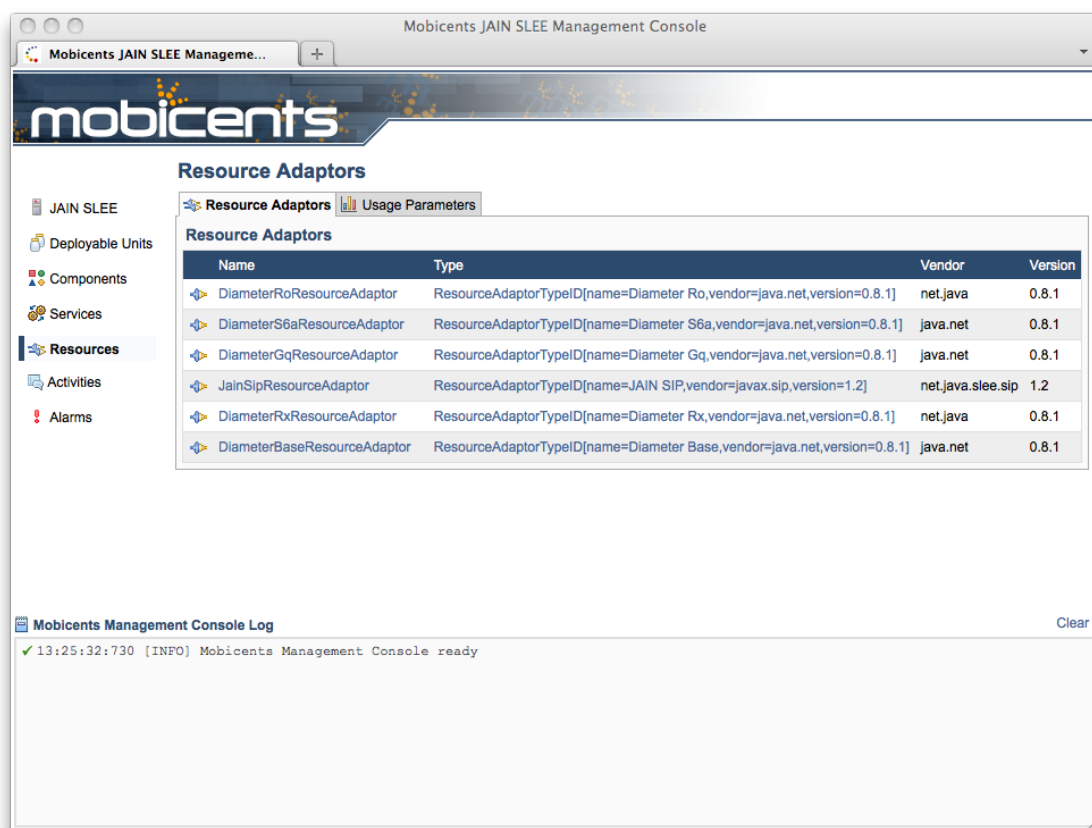


Figure 4.11. Resources Main View

4.5.1. View Resource Adaptor Details

In the **Resources** tab the list of Resource Adaptors is shown. When clicking on one of the resource adaptors' name, it's details are shown. The following details are displayed:

- Name

The SLEE component Name identifier.

- ID

The SLEE component identifier, built from the component type, name, vendor and version.

- Vendor

The SLEE component Vendor identifier.

- Version

The SLEE component Version identifier.

- Source

The path to the jar containing this component, inside the deployable unit jar.

- Deployable Unit

The deployable unit used to deploy this component.

- Library References

The libraries this component refers, if any.

- Resource Adaptor Entities

A list of the created Resource Adaptor Entities for this Resource Adaptor. These entities are clickable, and will lead to view the Entities details.

4.5.2. View and Change Resource Adaptor Entities State

When the existing Resource Adaptor Entities list is shown, the Resource Adaptor Entities state is presented in the **State** column, it can be:

- INACTIVE

The resource adaptor entity has been successfully created in the SLEE. An Inactive resource adaptor entity is ready to be activated.

In this state it is possible to activate the resource adaptor entity by clicking the **activate** link.

- ACTIVE

The resource adaptor entity has been activated. If the SLEE is in the Running state, resource adaptor objects associated with the resource adaptor entity can create new activities, submit events on activities, and end activities. If the SLEE is in the Stopping state, resource adaptor objects associated with the resource adaptor entity cannot start new activities, but may submit events on existing activities and end activities.

In this state it is possible to deactivate the resource adaptor entity by clicking the **deactivate** link.

- STOPPING

The resource adaptor entity is being deactivated. However, some activities created by the resource adaptor objects associated with the resource adaptor entity may still exist in the SLEE and have not completed their processing. The SLEE is waiting for these activities to end. Once all activities owned by resource adaptor objects of the resource adaptor entity have ended, the SLEE will transition the resource adaptor entity back to the Inactive state.

No state changing operations are available in this state, as this is a transition state, from ACTIVE to INACTIVE.

4.5.3. Create a Resource Adaptor Entity

In order to create a new Resource Adaptor Entity, after selecting the Resource Adaptor, fill the new entity name in the textbox next to **Create entity** and click the **Create** button. A new Entity will be created with default properties and in INACTIVE state.

4.5.4. Remove a Resource Adaptor Entity



Requirements to remove a Resource Adaptor Entity

To remove a Resource Adaptor Entity, there should be no Resource Adaptor Entity Links for this entity and it should be in `INACTIVE` state.

In order to remove a Resource Adaptor Entity, after selecting the Resource Adaptor, click the **remove** link. Make sure the above pre-requirements above are fulfilled.

4.5.5. View Resource Adaptor Entity Details

After selecting the Resource Adaptor, a list of the existent Resource Adaptor Entities can be found in the bottom of the panel. By clicking one of the Entities, the following details can be observed:

- Name

The Resource Adaptor Entity Name.

- State

The current state of the Resource Adaptor Entity.

- Entity Configuration Properties

The Resource Adaptor configuration properties provide a standardized mechanism for configuration of a resource adaptor entity. They consist of a triplet, Name, Type and Value.

- Entity Links

The Resource Adaptor Entity Links provide a link to an entity, to be used by SBBs.

4.5.6. View and Change Resource Adaptor Entity Configuration Properties

After selecting the Resource Adaptor Entity, a list of the Resource Adaptor Entity Configuration Properties is shown. They are displayed as `<name> :: <type> - <value>`. Only the value can be changed.



Requirements to edit Resource Adaptor Entity Configuration Properties

To edit the Resource Adaptor Entity, if the Resource Adaptor does not support reconfiguration in `ACTIVE` state, it must be in `INACTIVE` state.

To change these values, click the **Edit Properties**. The values will change to an editable form. Change the values accordingly and click the **Save** button.

4.5.7. View and Change Resource Adaptor Entity Links

After selecting the Resource Adaptor Entity, a list of the Resource Adaptor Entity Links is shown. It is possible to remove existing and add new links.



Requirements to remove a Resource Adaptor Entity Link

To remove a Resource Adaptor Entity Link, there should be no SBBs referring it.

To create a new Resource Adaptor Entity Link, fill the new link name in the **Bind entity link** textbox and click the **Bind** button.

To remove an existing Resource Adaptor Entity Link, click the **unbind** link.

4.5.8. Resource Adaptor Usage Parameters

In the **Usage Parameters** tab is possible to view and manage the Resource Adaptors' Entities Usage Parameters. In this screen, select a Resource Adaptor and a Resource Adaptor Entity and if a Resource Adaptor Usage Parameter Interface is defined, this view will present the default and other available Usage Parameter Sets.

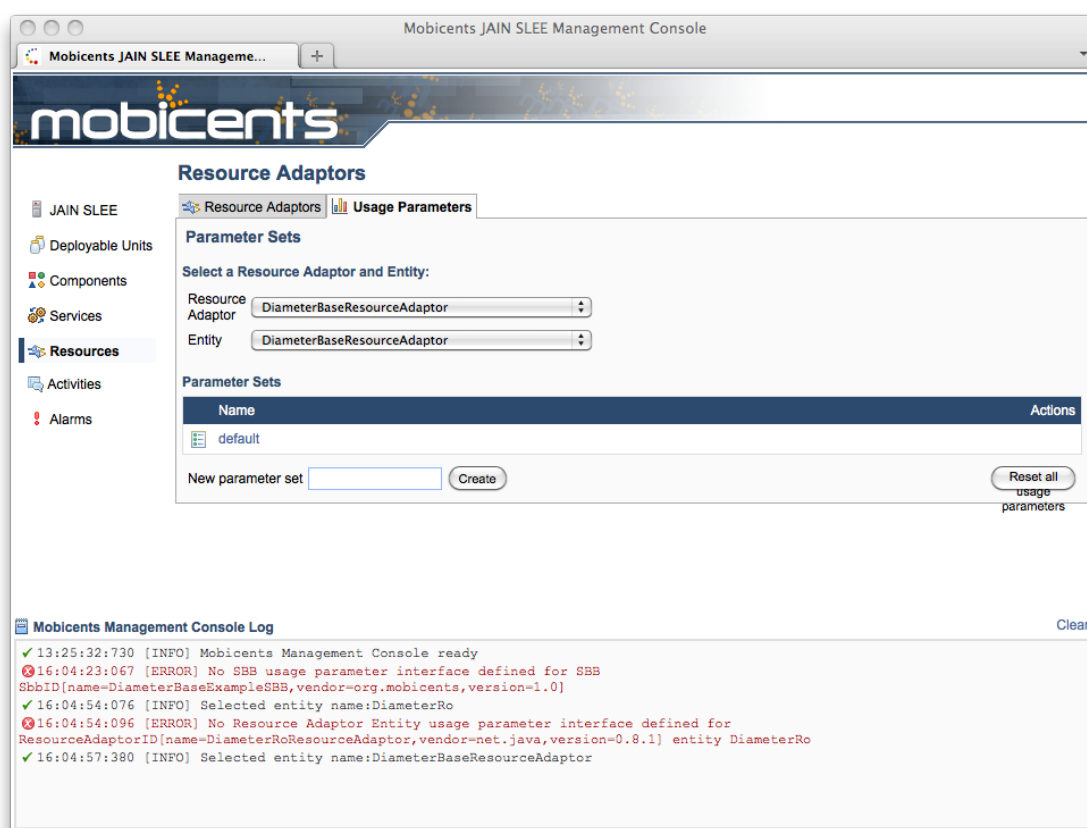


Figure 4.12. Resource Adaptor Usage Parameters View

4.5.8.1. View Resource Adaptor Usage Parameters

In order to view the Usage Parameters for a given set click it's name.

Resource Adaptors

Resource Adaptors | Usage Parameters

Parameter Sets → default

default

Resource Adaptor: DiameterBaseResourceAdaptor

Entity: DiameterBaseResourceAdaptor

Parameter set name: default

Counter type usage parameters [refresh](#)

Name	Value	Actions
FirstCount	0	reset
SecondCount	0	reset

Sample type usage parameters

Name	Minimum	Mean	Maximum	Sample count	Actions
TimeBetweenNewConnections	9223372036854775807	0	-9223372036854775808	0	reset
TimeBetweenErrors	9223372036854775807	0	-9223372036854775808	0	reset

[Reset all usage parameters](#)

Mobicents Management Console Log [Clear](#)

```

13:25:32:730 [INFO] Mobicents Management Console ready
16:04:23:067 [ERROR] No SBB usage parameter interface defined for SBB
SbbID[name=DiameterBaseExampleSBB,vendor=org.mobicents,version=1.0]
16:04:54:076 [INFO] Selected entity name:DiameterRo
16:04:54:096 [ERROR] No Resource Adaptor Entity usage parameter interface defined for
ResourceAdaptorID[name=DiameterRoResourceAdaptor,vendor=net.java,version=0.8.1] entity DiameterRo
16:04:57:380 [INFO] Selected entity name:DiameterBaseResourceAdaptor
  
```

Figure 4.13. Resource Adaptor Usage Parameters Details View

The following types are displayed:

- Counter Type

A counter-type usage parameter can be incremented or decremented. The current (approximate) value of the counter-type usage parameter is displayed under the **Value** column is shown.

- Sample Type

A sample-type usage parameter accumulates sample data, when a SLEE Component submit samples to the usage parameter. The current (approximate) minimum, maximum, mean, and the number of sample values added to the sample-type usage parameter are shown.



No automatic refresh

In order to keep a low impact on the SLEE no real-time update is performed with regards to the Usage Parameters. To update the displayed values, please click the **refresh** link.

4.5.8.2. Reset Resource Adaptor Usage Parameters

It is possible to reset a single Usage Parameter, when viewing the Resource Adaptor Usage Parameters list, by clicking the **reset** link, next to each, both for Counter and Sample type usage parameters.

In order to reset all the Usage Parameters for a given Resource Adaptor, in the main Usage Parameters view click the **Reset all usage parameters** button and all will be reset.

4.5.8.3. Create Named Resource Adaptor Usage Parameters Set

In addition to the default usage parameter set created by the SLEE, it is possible to create additional usage parameter sets for the same notification source, which must have a name. To create one, in the main Usage Parameters view, fill the **New parameter set** with the name and press the **Create** button and the new set is created.

4.5.8.4. Remove Named Resource Adaptor Usage Parameters Set

To remove a created Usage Parameter Set, in the main Usage Parameters view, under the **Parameter Sets** list, in the **Actions** column, click the **remove** link and the set will be removed.



The "default" Usage Parameters Set is not removable

Only named Usage Parameters Sets can be removed, the "default" will always exist when there's a Usage Parameter Interface defined for this component.

Monitoring Mobicents JAIN SLEE with Mobicents JAIN SLEE Management Console

5.1. "Activities" View

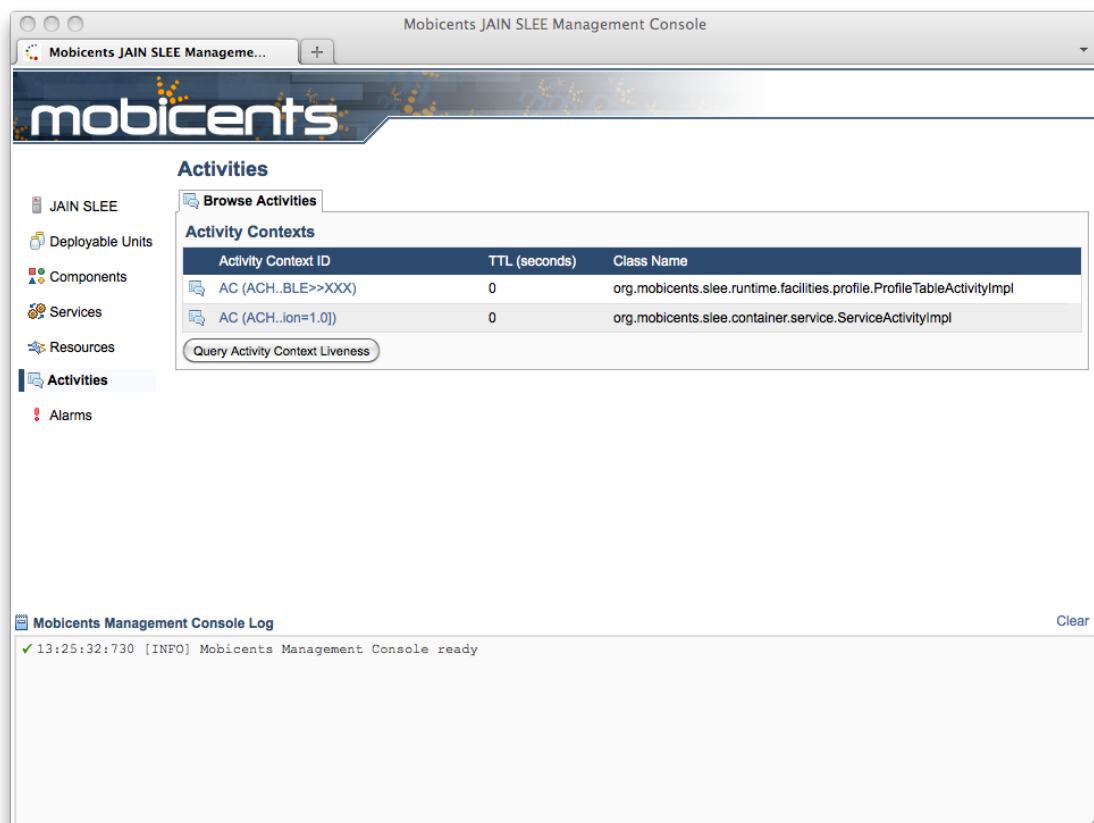


Figure 5.1. Activities Main View

In this view, it's possible to monitor the existing activities in the SLEE. The Activity Contexts are listed by their ID, TTL (maximum time in seconds without being accessed) and their Java Class Name.

5.1.1. View Activities Details

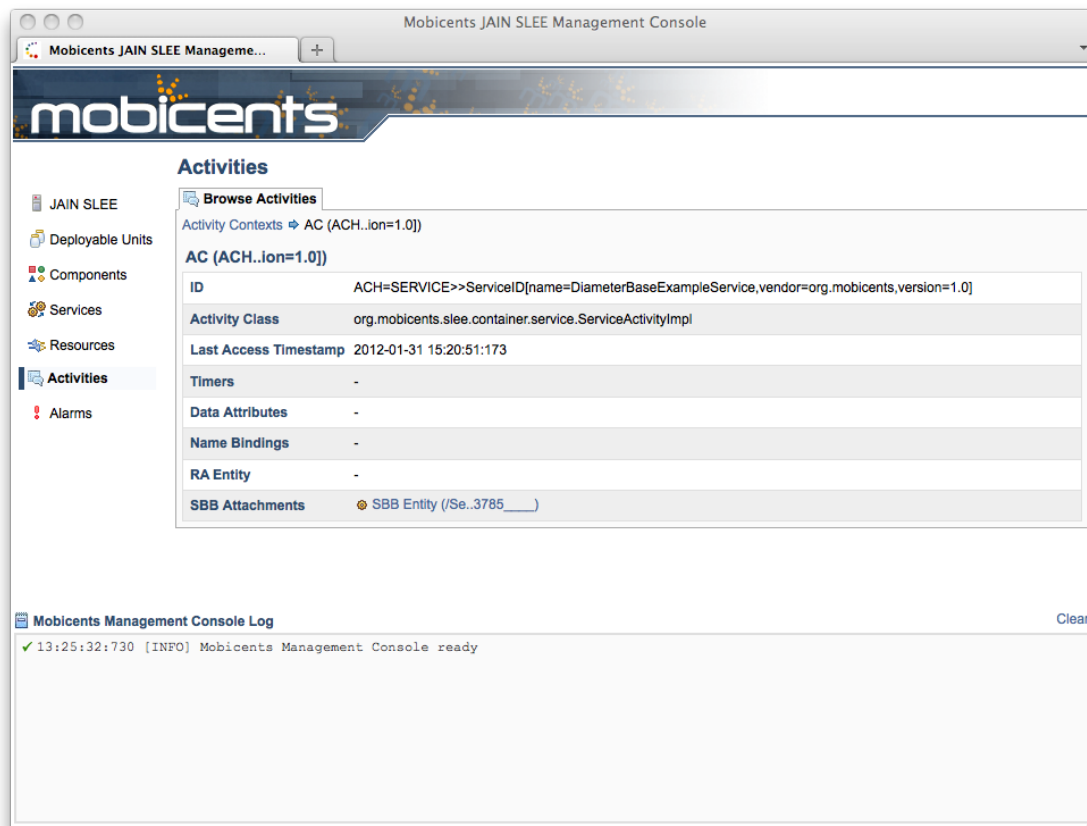


Figure 5.2. Activities Details View

In the **Browse Activities** tab the list of activities is shown. When clicking on one of the Activity Context ID, it's details are shown. The following details are displayed:

- ID

The internal identifier for this Activity Context.

- Activity Class

The Java class representing this activity.

- Last Access Timestamp

The timestamp of when this Activity Context was last accessed.

- Timers

The SLEE Timers attached to this Activity Context, if any.

- Data Attributes

The CMP attributes set in this Activity Context, if any.

- Name Bindings

The set of names given to this Activity Context, if any.

- RA Entity

The Resource Adaptor Entity responsible for creating this Activity Context, if any.

- SBB Attachments

The SBBs attached to this Activity Context, if any.

It is also possible to view the details of the SBB Entities attached to the activities by clicking on it's ID.

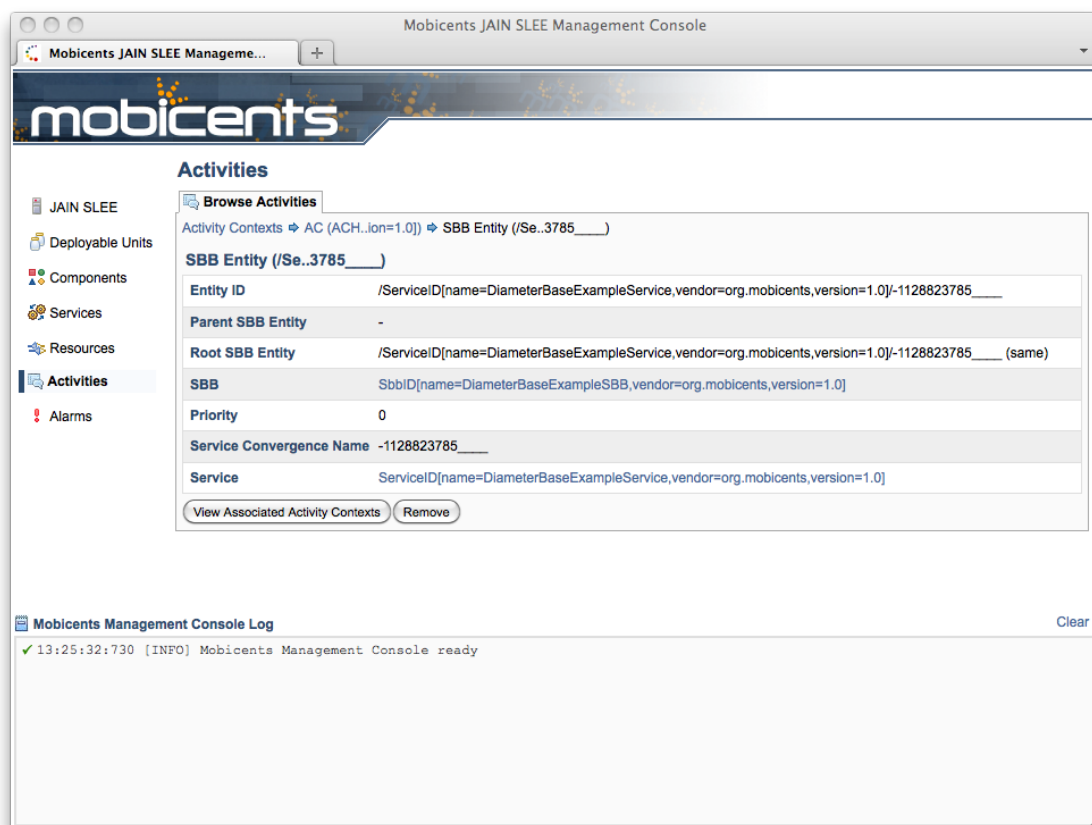


Figure 5.3. Activities SBB Entity Details View

The following details are shown:

- Entity ID

The internal identifier for this SBB Entity.

- Parent SBB Entity

The internal identifier for the parent SBB Entity for this SBB Entity, if any.

- Root SBB Entity

The internal identifier of the root SBB Entity for this SBB Entity, which may be itself.

- SBB

The SBB component this SBB Entity represents.

- Priority

The priority for this SBB Entity in it's child relation, if any.

- Service Convergence Name

A name used to uniquely identify a root SBB entity belonging to a Service.

- Service

The SLEE Service to which this SBB Entity belongs.

5.1.2. Query Activity Context Liveness

It is possible to query Resource Adaptor created activities for validity, by querying their liveness. If the Activity is not alive the Resource Adaptor is expected to end the activity. If the Activity is still alive the Resource Adaptor is not expected to do anything.

5.2. "Alarms" View

Alarms

! Active Alarms

Alarms

Total alarms :7 [refresh](#)

Timestamp	ID	Level	Message	Action
2012-01-31 15:15:20:631	7ba0a81d-5a69-41ff-a488-ce3e8cf6ef44	Major	HbH: 1001	clear
2012-01-31 15:15:21:592	17ab7b63-1dc3-42d3-8252-fa20006e670a	Warning	HbH: 1002	clear
2012-01-31 15:16:03:197	02ff0a3a-03d8-4f5d-b63c-9085d4d0e9a1	Critical	HbH: 1003	clear
2012-01-31 15:16:04:205	6c288890-bf4c-4f57-bae5-7857619d61eb	Major	HbH: 1004	clear
2012-01-31 15:16:05:212	added963d3-9d37-4719-bbf4-7c23fdde1b26	Major	HbH: 1005	clear
2012-01-31 15:16:06:219	3ef4ecf7-d87e-4827-bcce-100f3b501a30	Major	HbH: 1006	clear
2012-01-31 15:16:07:228	364c64f4-fb89-45ab-8ac1-8870deb28b88	Major	HbH: 1007	clear

Mobicents Management Console Log [Clear](#)

✓ 13:25:32:730 [INFO] Mobicents Management Console ready

Figure 5.4. Alarms Main View

In this view, it's possible to monitor the alarms which occurred in the SLEE. The alarms are listed by their Timestamp, ID, Level and Message.



No automatic refresh

In order to keep a low impact on the SLEE no real-time update is performed with regards to the Usage Parameters. To update the displayed values, please click the **refresh** link.

5.2.1. View Alarms Details

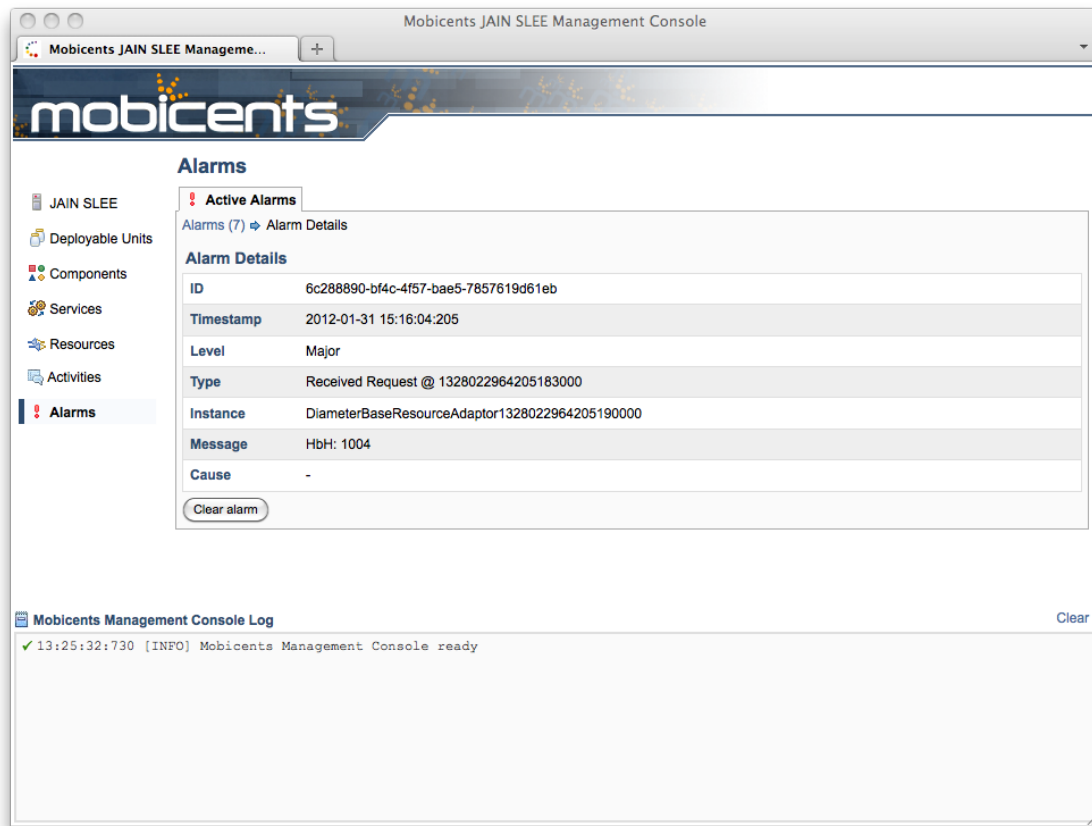


Figure 5.5. Alarms Details View

In the **Active Alarms** tab the list of alarms is shown. When clicking on one of the alarms ID, it's details are shown. The following details are displayed:

- ID

The internal identifier for this alarm.

- Timestamp

The Java class representing this activity.

- Level

The alarm level of the alarm represented by the alarm notification. In order of severity, from highest to least, they are: CLEAR, CRITICAL, MAJOR, WARNING, INDETERMINATE, MINOR.

- Type

The alarm type of the generated alarm, set by the alarm source.

- Instance

The instance of the generated alarm, set by the alarm source.

- Message

The message of the alarm notification.

- Cause

The Throwable which caused this alarm to be raised, if any, typically used to propagate an exception in the alarm notification.

5.2.2. Clear an Alarm

It is possible to clear an alarm by clicking the **clear** button in the list view or by clicking the **Clear alarm** button in the details view of an alarm.

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 higher¹.

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 <http://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 exported 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 session¹.

```
~]$ 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

Mon Jan 09 2012

AlexandreMendonça

Creation of the Mobicents JAIN SLEE Management Console User Guide.

Index

F

feedback, viii

