JBoss Communications JAIN SLEE SMPP5 Resource Adaptor User Guide

by Amit Bhayani

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 the the *Issue Tracker* [http://bugzilla.redhat.com/bugzilla/], against the product **JBoss Communications JAIN SLEE SMPP5 Resource Adaptor**, or contact the authors.

When submitting a bug report, be sure to mention the manual's identifier: JAIN_SLEE_SMPP5_RA_User_Guide

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 JBoss Communications JAIN SLEE SMPP5 Resource Adaptor

This resource adaptor provides a SMPP API for JAIN SLEE applications, adapting the SMPP specification version 5.0.

The Short Message Peer to Peer (SMPP) protocol is an open, industry standard protocol designed to provide a flexible data communications interface for transfer of short message data between a Message Center, such as a Short Message Service Centre (SMSC), GSM Unstructured Supplementary Services Data (USSD) Server or other type of Message Center and a SMS application system, such as a WAP Proxy Server, EMail Gateway or other Messaging Gateway.

Using the SMPP protocol, an SMS application system called the External Short Message Entity (ESME) may initiate an application layer connection with an SMSC over a TCP/IP network connection and may then send short messages and receive short messages to and from the SMSC respectively. The ESME may also query, cancel or replace short messages using SMPP.

SMPP supports a full featured set of two-way messaging functions such as:

- Transmit messages from an ESME to single or multiple destinations via the SMSC
- An ESME may receive messages via the SMSC from other SME's (e.g. mobile stations).
- Query the status of a short message stored on the SMSC
- Cancel or replace a short message stored on the SMSC
- Send a registered short message (for which a delivery receipt will be returned by the SMSC to the message originator)
- Schedule the message delivery date and time
- Select the message mode, i.e. datagram or store and forward
- Set the delivery priority of the short message
- Define the data coding type of the short message
- Set the short message validity period
- Associate a service type with each message e.g. voice mail notification
- Cell Broadcast Services

Events represent SMPP messages, or failure use cases such as timeouts.

The Activities are the SMPP Transaction, which applications in the SLEE may use to send SMPP message's and message responses, and to receive the events related with incoming message or message response.

Resource Adaptor Type

SMPP5 Resource Adaptor Type is defined by Mobicents team as part of effort to standardize RA Types.

2.1. Activities

An SMPP activity object represents a set of related events in an SMPP resource. This RA Type defines the following Activity objects:

SmppTransaction

The SmppTransaction represents the message submitted/received and corresponding message response received/submitted. The Activity begins when ResourceAdaptor receives a SMS Request and fires the corresponding event to SBB and ends as soon as listening SBB sends back Response.If there is no SBB listening for this event, the timeout occurs and Activity is killed. Other way is Activity is started as soon as SBB sends SMS message and ends once RA receives the message response or timeout occurs, which ever is first. Class name is `net.java.slee.resources.smpp.SmppTransaction`

New `SmppTransaction` Activity objects are created by calling `SmppSession.sendRequest(SmppRequest request)` This method is called by application that wants to send new SMS message.

`SmppTransaction` Activity objects are created automatically when the resource adaptor receives an incoming SMS message.

2.2. Events

Events represent SMPP message or response to message and Timer expiry. Each SMPP message and response is fired as different event types. Events are fired on `SmppTransaction` activity. Following is the table that describes event-type (name, vendor and version), event-class.

Important

Spaces where introduced in `Name` and `Event Class` column values, to correctly render the table. Please remove them when using copy/paste.

Important

For proper render of this table prefixes, for entries on some columns are ommited. For prefix values, for each column, please see list below:

Name

net.java.slee.resources.smpp.

Event Class
net.java.slee.resources.smpp.pdu.

Version for all defined events is 5.0

Vendor for all defined events is net.java.

Table 2.1. Events fired by SMPP5

Name	Event Class	Comments
ALERT_ NOTIFICATION	AlertNotification	This message is sent by the SMSC to the ESME, when the SMSC has detected that a particular mobile subscriber has become available and a delivery pending flag had been set for that subscriber from a previous data_sm operation.
GENERIC_NACK	GenericNack	This is a generic negative acknowledgement to an SMPP PDU submitted with an invalid message header.
DELIVER_SM	DeliverSM	The deliver_sm is issued by the SMSC to send a message to an ESME. Using this command, the SMSC may route a short message to the ESME for delivery.
DELIVERY_REPORT	DeliverSM	SMSC Delivery Receipt. A delivery receipt relating to a a message which had been previously submitted with the submit_sm operation and the ESME had requested a delivery receipt via the registered_delivery parameter. The delivery receipt data relating to the original short message will be included in the short_message field of the deliver_sm.
DELIVER_SM _RESP	DeliverSMResp	The SMPP PDU response sent from an ESME that received DELIVER_SM message
SUBMIT_SM	SubmitSM	This operation is used by an ESME to submit a short message to the SMSC for onward transmission to a specified short message entity (SME).

Name	Event Class	Comments
SUBMIT_SM _RESP	SubmitSMResp	This is the response to the submit_sm PDU
DATA_SM	DataSM	This command is used to transfer data between the SMSC and the ESME. It may be used by both the ESME and SMSC. This command is an alternative to the submit_sm and deliver_sm commands. It is introduced as a new command to be used by interactive applications such as those provided via a WAP framework.
DATA_SM _RESP	DataSMResp	This is the response to the data_sm PDU
SUBMIT_MULTI	SubmitMulti	The submit_multi operation may be used to submit an SMPP message for delivery to multiple recipients or to one or more Distribution Lists
SUBMIT_MULTI _RESP	SubmitMultiResp	This is the response to the submit_multi PDU
QUERY_SM	QuerySM	This command is issued by the ESME to query the status of a previously submitted short message.
QUERY_SM _RESP	QuerySMResp	This is the response to the query_sm PDU
CANCEL_SM	CancelSM	This command is issued by the ESME to cancel one or more previously submitted short messages that are still pending delivery.
CANCEL_SM _RESP	CancelSMResp	This is the response to the cancel_sm PDU
REPLACE_SM	ReplaceSM	This command is issued by the ESME to replace a previously submitted short message that is still pending delivery.
REPLACE_SM _RESP	ReplaceSMResp	This is the response to the replace_sm PDU
BROADCAST_SM	BroadcastSM	This operation is issued by the ESME to submit a message to the Message Centre for broadcast to a specified geographical area or set of geographical areas.

Name	Event Class	Comments
BROADCAST_SM_RESP	BroadcastSMResp	This is the response to the broadcast_sm PDU
QUERY_BROADCAST_SM	QueryBroadcastSM	This command is issued by the ESME to query the status of a previously submitted broadcast message
QUERY_BROADCAST_SM_RESP	QueryBroadcastSMResp	This is the response to the query_broadcast_sm PDU
CANCEL_BROADCAST_SM	CancelBroadcastSM	This command is issued by the ESME to cancel a broadcast message which has been previously submitted to the Message Centre for broadcast via broadcast_sm and which is still pending delivery.
CANCEL_BROADCAST_SM_RESP	CancelBroadcastSMResp	This is the response to the cancel_broadcast_sm PDU
SMPP_TIMEOUT_RESPONSE_SENT	SmppError	This is issued when ESME receives SMPP PDU and doesn't send the corresponding response in configured time limit
SMPP_TIMEOUT_RESPONSE_RECEIVED	SmppError	This is issued when ESME has sent SMPP PDU but there is no corresponding response from SMSC in configured time limit

2.3. Activity Context Interface Factory

The interface of the SMPP resource adaptor type specific Activity Context Interface Factory is defined as follows:

```
package net.java.slee.resources.smpp;

public interface SmppTransactionACIFactory {
 javax.slee.ActivityContextInterface getActivityContextInterface(SmppTransaction txn);

}
```

2.4. Resource Adaptor Interface

The SMPP Resource Adaptor SBB Interface facilitates SBBs to create new SMS message and send it. It also facilitates sending of SMS message response. The isAlive() is check for connection between RA and SMSC is still alive. It is defined as follows:

```
package net.java.slee.resources.smpp;

import net.java.slee.resources.smpp.pdu.Address;
import net.java.slee.resources.smpp.pdu.SmppRequest;
import net.java.slee.resources.smpp.pdu.SmppResponse;

public interface SmppSession {

 public String getSessionId();

 public String getSMSCHost();

 public int getSMSPort();

 public SmppTransaction sendRequest(SmppRequest request) throws
 java.lang.IllegalStateException, java.lang.NullPointerException,
 java.io.IOException;

 public void sendResponse(SmppTransaction txn, SmppResponse
 response) throws java.lang.IllegalStateException,
 java.lang.NullPointerException, java.io.IOException;

 public boolean isAlive();

 public SmppRequest createSmppRequest(long commandId);

 public Address createAddress(int addTon, int addNpi, String
 address);

}
```

2.5. Sbb Code Examples

2.5.1. Accesing resource adaptor from Sbb

Sbb explicitly lookup resource adaptor object using resource-adaptor-entity-binding element in the SBB's deployment descriptor

```
SmppTransactionACIFactory smppAcif;
SmppSession smppSession;

public void setSbbContext(SbbContext sbbContext) {
        this.sbbContext = sbbContext;
        try {
            logger.info("Called setSbbContext PtinAudioConf!!!");
            Context myEnv = (Context) new InitialContext().lookup("java:comp/env");

            smppSession = (SmppSession)
                    myEnv.lookup("slee/resources/smpp/5.0/smppSession");
            smppAcif = (SmppTransactionACIFactory)
                    myEnv.lookup("slee/resources/smpp/5.0/factoryprovider");
        } catch (NamingException ne) {
            logger.warn("Could not set SBB context:" + ne.getMessage());
        }
}
```

2.5.2. Handling Message from SMSC

Each time message received from SMSC, the resource adaptor emits DELIVER_SM message. The DeliverSM event object is used to create the corresponding response DeliverSMResp passing the command_status. The SmppTransaction can be retrieved from ActivityContextInterface object. The SmppSession retrieved from SmppTransaction is used to send back the response. Once the response is sent, the Activity dies.

```
    public void onSmsMessage(DeliverSM event, ActivityContextInterface aci) {
            ....
          DeliverSMResp resp = event.createSmppResponseEvent(SmppTransaction.ESME_ROK);
            SmppTransaction txn = (SmppTransaction)aci.getActivity();
            txn.getSmppSession().sendResponse(txn, response);
    }
```

2.5.3. Submitting message from SBB

The SmppSession object should be used to prepare outgoing message and a SmppTransaction.

```
   public void onSomeEvent(SomeEvent event, ActivityContextInterface aci) {
                                                                SubmitSM submitSm =
(SubmitSM)smppSession.createSmppRequest(SmppRequest.SUBMIT_SM);

       Address esmeAddress = smppSession.createAddress(1, 0, "501");
       Address destAddress = smppSession.createAddress(1, 0, "919960666666");

       submitSm.setEsmeAddress(esmeAddress);
       submitSm.setEsmeAddress(destAddress);
       submitSm.setMessage("Hello World".getBytes());

       SmppTransaction submitTxn = smppSession.sendRequest(submitSm);

       // attach to the new activity so we get the response
       ActivityContextInterface newaci = smppAcif.getActivityContextInterface(submitTxn);
       newaci.attach(getSbbLocalObject());
}
```

Resource Adaptor Implementation

The RA implementation uses the SMPP Impl from *smppapi* [http://smppapi.sourceforge.net/documentation/user-guide/] poject.

3.1. Configuration

The Resource Adaptor supports configuration only at Resource Adaptor Entity creation time, the following table enumerates the configuration properties:

Table 3.1. Resource Adaptor's Configuration Properties

Property Name	Description	Property Type	Default Value
host	The host address of SMSC that this RA Entity should connect to	java.lang.String	localhost
port	The Port of SMSC that this RA Entity should connect to	java.lang.Integer	2775
systemId	The System ID field for bind request. Used by the SMSC for authentication.	java.lang.String	smppclient1
systemType	Identifies the type of ESME system requesting to bind as a receiver/transmitter/transciever with the SMSC.	java.lang.String	ESME
password	Password field for bind request. Used by the SMSC for authentication.	java.lang.String	password
addressTon	The type of Address for this ESME, used in BIND request	java.lang.Integer	1
addressNpi	The Number Plan Indicator for this ESME, used in BIND request	java.lang.Integer	0

Property Name	Description	Property Type	Default Value
addressRange	A single ESME address or a range of ESME addresses served via this SMPP receiver/ transciever/ transmitter session. The parameter value is represented in UNIX regular expression format (see Appendix A) of SMPP Spec 3.4. Set to NULL if not known.	java.lang.Integer	50
enquireLinkTimeout	Interval (in seconds) between the RA entity sending enquire_link requests to test the SMPP session is still live.	java.lang.Integer	30
smppResponseReceivedTimeout	Maximum time (in ms) to wait for a response to an outgoing SMPP request. If this timer fires, a SmppErrorEvent is sent to the application, containing the SMPP request PDU that failed.	java.lang.Integer	5000
smppResponseSentTimeout	Maximum time (in ms) to wait for the application (SBB) to respond to an incoming SMPP request. If this timer fires, a generic_nack PDU is sent to the other node, with status ESME_RSYSERR,	java.lang.Integer	5000

Property Name	Description	Property Type	Default Value
	and the activity is ended.		

Important

JAIN SLEE 1.1 Specification requires values set for properties without a default value, which means the configuration for those properties are mandatory, otherwise the Resource Adaptor Entity creation will fail!

3.2. Default Resource Adaptor Entities

There is a single Resource Adaptor Entity created when deploying the Resource Adaptor, named SmppRA. The SmppRA entity uses the default Resource Adaptor configuration, specified in *Section 3.1, "Configuration"*. The SmppRA entity is also bound to Resource Adaptor Link Name SmppRA, to use it in an Sbb add the following XML to its descriptor:.

```
    <resource-adaptor-type-binding>
      <resource-adaptor-type-ref>
        <resource-adaptor-type-name>SMPPResourceAdaptorType</resource-adaptor-type-name>
        <resource-adaptor-type-vendor>net.java</resource-adaptor-type-vendor>
        <resource-adaptor-type-version>5.0</resource-adaptor-type-version>
      </resource-adaptor-type-ref>
     <activity-context-interface-factory-name>slee/resources/smpp/5.0/factoryprovider</activity-context-interface-factory-name>
      <resource-adaptor-entity-binding>
        <resource-adaptor-object-name>slee/resources/smpp/5.0/smppinterface</resource-adaptor-object-name>
        <resource-adaptor-entity-link>SmppRA</resource-adaptor-entity-link>
      </resource-adaptor-entity-binding>
    </resource-adaptor-type-binding>
```

3.3. Traces and Alarms

3.3.1. Tracers

Each Resource Adaptor Entity uses a single JAIN SLEE 1.1 Tracer, named SmppResourceAdaptor.

3.3.2. Alarms

The RA raises an alarm identified as `SMSCALARM` if for some reason the link with SMSC is broken. This alarm is cleared as soon as the RA is able to establish the link again or if management clears the alarm explicitly.

Setup

4.1. Pre-Install Requirements and Prerequisites

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

4.1.1. Hardware Requirements

The Resource Adaptor hardware's main concern is RAM memory and Java Heap size, the more the better. For instance, while the underlying JBoss Communications JAIN SLEE may run with 1GB of RAM.

Of course, memory is only needed to store the Resource Adaptor state, the faster the CPU more transactions per second are supported, yet no particular CPU is a real requirement to use the RA.

4.1.2. Software Prerequisites

The RA requires JBoss Communications JAIN SLEE properly set.

4.2. JBoss Communications JAIN SLEE SMPP5 Resource Adaptor Source Code

4.2.1. Release Source Code Building

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, lets consider 1.0.0.FINAL.

```
[usr]$ svn co ?/1.0.0.FINAL slee-ra-smpp5-1.0.0.FINAL
```

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*

Use Maven to build the deployable unit binary.

```
[usr]$ cd slee-ra-smpp5-1.0.0.FINAL
[usr]$ mvn install
```

Once the process finishes you should have the `deployable-unit` jar file in the `target` directory, if JBoss Communications JAIN SLEE is installed and environment variable JBOSS_HOME is pointing to its underlying JBoss Enterprise Application Platform directory, then the deployable unit jar will also be deployed in the container.

4.2.2. Development Trunk Source Building

Similar process as for *Section 4.2.1, "Release Source Code Building"* , the only change is the SVN source code URL, which is NOT AVAILABLE.

4.3. Installing JBoss Communications JAIN SLEE SMPP5 Resource Adaptor

To install the Resource Adaptor simply execute provided ant script `build.xml` default target:

```
[usr]$ ant
```

The script will copy the RA deployable unit jar to the `default` JBoss Communications JAIN SLEE server profile deploy directory, to deploy to another server profile use the argument `-Dnode=` .

4.4. Uninstalling JBoss Communications JAIN SLEE SMPP5 Resource Adaptor

To uninstall the Resource Adaptor simply execute provided ant script `build.xml undeploy` target:

```
[usr]$ ant undeploy
```

The script will delete the RA deployable unit jar from the `default` JBoss Communications JAIN SLEE server profile deploy directory, to undeploy from another server profile use the argument `-Dnode=` .

Clustering

5.1. Failover

SMPP RA doesn't replicate the SmppTransaction activity for failover. This is due to inherent nature of SMPP Protocol which is peer-to-peer

5.2. Load Balancing

SMPP RA implements the HA API (FaultTolerantResourceAdaptor) of Mobicents JSLEE. Same entity of SMPP RA can be started in different cluster enabled nodes of JSLEE to take care of load balancing.

Appendix A. Revision History

Revision History
Revision 1.0 Wed Sep 22 2010 AmitBhayani
Creation of the JBoss Communications JAIN SLEE SMPP5 RA User Guide.

Index

F