User Guide to Restcomm jSS7 ISUP RA 8.0.0-SNAPSHOT

Table of Contents

Preface

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 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.

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 to switch to the first virtual terminal. Press 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 menu:>[] shorthand used to indicate traversal through a menu and its sub-menus. This is to avoid the difficult-to-follow 'Select from the **Preferences** | **] sub-menu in the menu: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.

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"));
   }

}
```

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.

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 {this-issue.tracker.ur}, against the product Restcomm jSS7` `, or contact the authors.

When submitting a bug report, be sure to mention the manual's identifier: Restcomm jSS7

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.

Chapter 1. Introduction to Restcomm JAIN SLEE ISUP Resource Adaptor

The ISUP - User Part or ISUP - is part of the Signaling System #7 which is used to set up telephone calls in Public Switched Telephone Networks. It is specified by the ITU-T as part of the Q.76x series.

ISUP messages are used to convey call related information between PSTN switches.

The telephone switches are connected via E1 or T1 trunks which transport the speech from the calls. Trunks are divided into 64k/bit timeslots. Each time slot is identified with unique Circuit Identification Code (CIC). The switches uses this information along with the received signalling information (Called Party Number, Calling Part Number...) to determine which inbound CICs and outbound CICs should be connected together to provide an end to end speech path.

Additionally ISUP is used to exchange status information about switch capabilities and current resources.

Chapter 2. Resource Adaptor Type

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

2.1. Activities

ISUP activities object represents a set of related events in an ISUP resource.

This ISUP Type defines two types of activities:

ISUPClientTransaction

This activity is associated with client side events. All types of request are sent by means of this activity. Answers are delivered on this activity. Life cycle of this activity depends on message type. Client transaction stays alive until it timeouts or answer is delivered. Exception to this rule are messages which are only indication of state - they dont require any answer. This group consists of:

- Reset Circuit: Full class name is: `org.mobicents.protocols.ss7.isup.ISUPClientTransaction`. It is created on demand with RA SBB Interface.

ISUPServerTransaction

This activity is associated with server side events. All types of request are fired by on this activity. Answers are sent through this activity. Life cycle of this activity depends on message type. Server transaction stays alive until it timeouts or answer is sent back. Exception to this rule are messages which are only indication of state - they dont require any answer. See above activity. Full class name is: `org.mobicents.protocols.ss7.isup.ISUPServerTransaction`. It is created by RA for each request which does not have transaction associated with it.

2.2. Events

Events represent telephony switch actions taken over particular circuit.

Based on type events are fired on of two activities this Type defines.

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.resource.isup.

Event Class

org.mobicents.protocols.ss7.isup.message.

Version for all defined events is 1.0

Vendor for all defined events is org.mobicents

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

`Server` column indicates if event is received on server activity.

Table 1. Events fired by &THIS.RA;

Name	Event Class	Comments	Server
ADDRESS_COMPLETE	AddressCompleteMessage		Y
ANSWER	AnswerMessage		Y
APPLICATION_TRANSPORT	ApplicationTransportMessage		
BLOCKING	BlockingMessage		Y
BLOCKING_ACK	BlockingAckMessage		-
CALL_PROGRESS	CallProgressMessage		Y
CHARGE_INFORMATION	ChargeInformationMessage		Y
CIRCUIT_GROUP_BLOCKING	CircuitGroupBlockingMessage		Y
CIRCUIT_GROUP_BLOCKING_ACK	CircuitGroupBlockingAckMessage		N
CIRCUIT_GROUP_QUERY	CircuitGroupQueryMessage		Y
CIRCUIT_GROUP_QUERY_RESPONSE	CircuitGroupQueryResponseMessage		N
CIRCUIT_GROUP_RESET	CircuitGroupResetMessage		Y
CIRCUIT_GROUP_RESET_ACK	CircuitGroupResetAckMessage		N

Name	Event Class	Comments	Server
CIRCUIT_GROUP _UNBLOCKING	CircuitGroup UnblockingMessage		Y
CIRCUIT_GROUP _UNBLOCKING_ACK	CircuitGroup UnblockingAckMessage		N
CONNECT	ConnectMessage		Y
CONTINUITY_ CHECK_REQUEST	ContinuityCheck RequestMessage		Y
CONTINUITY	ContinuityMessage		N
FACILITY_ACCPETED	FacilityAcceptedMessag e		Z
FACILITY_REJECTED	FacilityRejectedMessag e		Z
FORWARD_TRANSFER	ForwardTransferMessa ge		Z
IDENTIFICATION_REQ UEST	Identification RequestMessage		Z
IDENTIFICATION_RESP ONSE	Identification ResponseMessage		Z
INITIAL_ ADDRESS_MESSAGE	InitialAddress Message		Y
LOOP_PREVENTION	LoopPrevention Message		Z
LOOPBACK_ACK	LoopbackAckMessage		Z
NETWORK_RESOURCE _MANAGEMENT	NetworkResource ManagementMessage		Z
OVERLOAD	OverloadMessage		Z
PASS_ALONG	PassAlongMessage		Z
PRERELEASE_INFORM ATION	PreRelease InformationMessage		Z
RELEASE	ReleaseMessage		Y
RELEASE_COMPLETE	ReleaseCompleteMessa ge		N
RESET_CIRCUIT	ResetCircuitMessage		Y
RESUME	ResumeMessage		Z
SUBSEQUENT_ADDRES S	SubsequentAddressMes sage		Z

Name	Event Class	Comments	Server
SUBSEQUENT_ DIRECTORY_NUMBER	SubsequentDirectory NumberMessage		Z
SUSPEND	SuspendMessage		Z
UNBLOCKING	UnblockingMessage		Y
UNBLOCKING_ACK	Unblocking AckMessage		N
UNEQUIPPED_CIC	Unequipped CICMessage		Z
USER_TO_ USER_INFORMATION	User2User InformationMessage		Z
USER_PART _AVAILABLE	UserPart AvailableMessage		Z
USER_PART_TEST	UserPart TestMessage		Z

There is one additional message defined. It is fired on both types of activities. It indicates timeout of transaction:

- Event name: TRANSACTION_ENDED
- Class name: org.mobicents.slee.resources.ss7.isup.events.TransactionEnded

2.3. Activity Context Interface Factory

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

```
package org.mobicents.slee.resources.ss7.isup.ratype;

import javax.slee.ActivityContextInterface;
import javax.slee.FactoryException;
import javax.slee.UnrecognizedActivityException;
import org.mobicents.protocols.ss7.isup.ISUPClientTransaction;
import org.mobicents.protocols.ss7.isup.ISUPServerTransaction;

public interface ActivityContextInterfaceFactory{

    /**
     * Gets ActivityContextInterface for client transaction activity.
     *
     * @param activity
     *  the endpoint activity object.
     * @return the ActivityContextInterface.
     */
    public ActivityContextInterface getActivityContextInterface(ISUPClientTransaction
activity)
        throws NullPointerException, UnrecognizedActivityException, FactoryException;

    /**
     * Gets ActivityContextInterface for server transaction activity.
     *
     * @param activity
     *  the endpoint activity object.
     * @return the ActivityContextInterface.
     */
    public ActivityContextInterface getActivityContextInterface(ISUPServerTransaction
activity)
        throws NullPointerException, UnrecognizedActivityException, FactoryException;

}
```

2.4. Resource Adaptor Interface

The ISUP Resource Adaptor Interface provides s with access to the factories required to create messages and parameters. It also provides means of creating activities:

```
package org.mobicents.slee.resources.ss7.isup.ratype;

import java.io.IOException;

import javax.slee.SLEEException;
import javax.slee.resource.ActivityAlreadyExistsException;

import javax.slee.resource.StartActivityException;
```

```
import org.mobicents.protocols.ss7.isup.ISUPClientTransaction;
import org.mobicents.protocols.ss7.isup.ISUPMessageFactory;
import org.mobicents.protocols.ss7.isup.ISUPParameterFactory;
import org.mobicents.protocols.ss7.isup.ISUPServerTransaction;
import org.mobicents.protocols.ss7.isup.ParameterRangeInvalidException;
import org.mobicents.protocols.ss7.isup.TransactionAlredyExistsException;
import org.mobicents.protocols.ss7.isup.message.ISUPMessage;

public interface RAISUPProvider {

    /**
     * Create client transaction activity
     * @param arg0
     * @return
     * @throws TransactionAlredyExistsException
     * @throws IllegalArgumentException
     * @throws ActivityAlreadyExistsException
     * @throws NullPointerException
     * @throws IllegalStateException
     * @throws SLEEException
     * @throws StartActivityException
     */
    public ISUPClientTransaction createClientTransaction(ISUPMessage arg0)
    throws TransactionAlredyExistsException,IllegalArgumentException,
    ActivityAlreadyExistsException, NullPointerException, IllegalStateException,
    SLEEException,  StartActivityException;
    /**
     * Create server transaction activity
     * @param arg0
     * @return
     * @throws TransactionAlredyExistsException
     * @throws IllegalArgumentException
     * @throws ActivityAlreadyExistsException
     * @throws NullPointerException
     * @throws IllegalStateException
     * @throws SLEEException
     * @throws StartActivityException
     */
    public ISUPServerTransaction createServerTransaction(ISUPMessage arg0)
    throws TransactionAlredyExistsException,IllegalArgumentException,
    ActivityAlreadyExistsException, NullPointerException, IllegalStateException,
    SLEEException, StartActivityException;

    /**
     * Get message factory.
     * @return
     */
    public ISUPMessageFactory getMessageFactory();
    /**
     * Get parameter factory.
     * @return
```

```
     */
    public ISUPParameterFactory getParameterFactory();
    /**
     * Send message statelesly.
     * @param arg0
     * @throws ParameterRangeInvalidException
     * @throws IOException
     */
    public void sendMessage(ISUPMessage arg0) throws ParameterRangeInvalidException,
IOException;
    /**
     * Determine if transport layer is connected and links are up.
     * @return
     */
    public boolean isTransportUp();
}
```

2.5. Restrictions

There are no known restrictions.

2.6. Sbb Code Examples

TODO

Chapter 3. Resource Adaptor Implementation

The RA implementation uses the Restcomm ISUP stack. The stack is the result of the work done by Restcomm JSLEE Server development teams, and source code is provided in all releases.

3.1. Configuration

The Resource Adaptor supports configuration only at Resource Adaptor Entity creation time. It supports following properties:

Table 2. Resource Adaptor's Configuration Properties - cap-default-ra.properties

Property Name	Description	Property Type	Default Value
mtp.driver	Configures driver of MTP layer	java.lang.String	m3ua
mtp.address.remote	Address of remote end of data link. It expects data in format: IP:Port .	java.lang.String	
mtp.address.local	As above. It points to local address to which data link is bound.	java.lang.String	
mtp.apc	Indicates adjacent point code (dpc for originating messages).	java.lang.Integer	
mtp.opc	Indiactes oritinating point code.	java.lang.Integer	
isup.client.timeout	Value of timeout in milisecond. It controls timeout of client transaction. This value must be lower than isup.general.timeout	java.lang.Long	30.000
isup.general.timeout	Value of timeout in millisecond. It controls how long transaction object lingers in stack before its released - in case no action is performed.	java.lang.Long	120.000

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 `ISUPResourceAdaptor`. The `ISUPResourceAdaptor` entity uses the default Resource Adaptor configuration, specified in Configuration.

The `ISUPResourceAdaptor` entity is also bound to Resource Adaptor Link Name `ISUPResourceAdaptor`, 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>
                    ISUPResourceAdaptorType
                </resource-adaptor-type-name>
                <resource-adaptor-type-vendor>
                    net.java
                </resource-adaptor-type-vendor>
                <resource-adaptor-type-version>
                    1.0
                </resource-adaptor-type-version>
            </resource-adaptor-type-ref>
            <activity-context-interface-factory-name>
                slee/resources/isup/1.0/acifactory
            </activity-context-interface-factory-name>
            <resource-adaptor-entity-binding>
            <resource-adaptor-object-name>
                    slee/resources/isup/1.0/provider
            </resource-adaptor-object-name>
                <resource-adaptor-entity-link>
                    ISUPResourceAdaptor
                </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 `ISUPResourceAdaptor`. The related Log4j Logger category, which can be used to change the Tracer level from Log4j configuration, is `javax.slee.RAEntityNotification[entity=IsupResourceAdaptor]`

3.3.2. Alarms

No alarms are set by this Resource Adaptor.

Chapter 4. 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.

Of course, memory is only needed to store the Resource Adaptor state, the faster the CPU more ISUP Messages processing is supported, yet no particular CPU is a real requirement to use the RA.

4.1.2. Software Prerequisites

The RA requires Restcomm JAIN SLEE properly set.

4.2. Restcomm JAIN SLEE ISUP Resource Adaptor Source Code

4.2.1. Release Source Code Building

1. Downloading the source code

Git is used to manage Restcomm JAIN SLEE source code. Instructions for downloading, installing and using Git can be found at http://git-scm.com/

Use Git to checkout a specific release source, the Git repository URL is https://github.com/Restcomm/jain-slee.ss7, then switch to the specific release version, lets consider 8.0.0-SNAPSHOT.

```
[usr]$ git clone https://github.com/Restcomm/jain-slee.ss7/
[usr]$ cd jain-slee.ss7
[usr]$ git checkout tags/{project-version}
```

2. Building the source code

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 resources/ISUP
[usr]$ mvn install
```

Once the process finishes you should have the `deployable-unit` jar file in the *target* directory, if Restcomm JAIN SLEE is installed and environment variable JBOSS_HOME is pointing to its underlying {jee.platform} directory, then the deployable unit jar will also be deployed in the container.

4.2.2. Development Master Source Building

Similar process as for Release Source Code Building, the only change is the Git reference should be the `master`. The `git checkout tags/8.0.0-SNAPSHOT` command should not be performed. If already performed, the following should be used in order to switch back to the master:

```
[usr]$ git checkout master
```

4.3. Installing Restcomm JAIN SLEE ISUP 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` Restcomm JAIN SLEE server profile deploy directory, to deploy to another server profile use the argument `-Dnode=`.

4.4. Uninstalling Restcomm JAIN SLEE ISUP 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` Restcomm JAIN SLEE server profile deploy directory, to undeploy from another server profile use the argument `-Dnode=`.

Chapter 5. Clustering

The RA implementation does not support clustering in current version.

Appendix A: Revision History