User Guide to Restcomm GMLC

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 GMLC` `, or contact the authors.

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

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 GMLC

Restcomm GMLC is an Open Source Java based GMLC Gateway Platform that routes ATI messages from the signaling network to service applications and the other way around. It enables operators to offer real-time interactions to mobile subscribers and deliver interactive content to their mobile phones. Restcomm GMLC strictly adheres to the standards and specifications defined by the International Telecommunications Union (ITU).

Restcomm GMLC acts as an intermediary platform linking the service applications to the GSM network in a session oriented communication. The platform is easy-to-install and easy-to-deploy allowing you to have the Gateway set up and configured very quickly.

Restcomm GMLC comes with an efficient Command Line Interface (CLI) tool allowing you to completely configure the Gateway at run-time and manage it using simple commands rather than do everything manually. Restcomm GMLC also comes with a Graphical User Interface (GUI) that will allow you to configure, monitor and manage the Gateway through a convenient user-friendly interface.

The Open Source Software gives you the flexibility to understand the readily available source code and the freedom to customise the product to meet your Enterprise needs.

This guide provides details on configuring and using the platform and information regarding the supported protocols and compliant standards. For installation instructions, please refer to the Installation Guide published along with this.

Chapter 2. Overview

2.1. GMLC

GMLC stands for Gateway Mobile Location Centre enables you to offer Location Based Services (LBS) to mobile subscribers in GSM and UMTS network.

In one PLMN (Public Land Mobile Network), there may be more than one GMLC. The GMLC is the first node an external LCS client accesses in a GSM or UMTS network. The GMLC may request routing information from the HLR (Home Location register) or HSS (Home Subscriber Server). After performing registration authorization, it sends positioning requests to either the VMSC (Visited Mobile Switching Centre), SGSN (Serving GPRS Support Node) or MSC (Mobile Switching Centre) Server and receives final location estimates from the corresponding entity.

2.2. GMLC Session

GMLC establishes a real time session between the mobile handset and the application handling the service and the information from the mobile handset is sent directly to the service. The concept of a real time session is very useful for constructing an interactive menu driven application. Refer to the figure below depicting the working of a real time session.

A user dialing an GMLC service number (short code) initiates a dialog with the GMLC handling application deployed on the Restcomm Platform as shown in the above figure. The "Network Node" in the figure could be a MSC, HLR or VLR. The Restcomm Platform integrates with the "Network Node" using the MAP protocol.

2.3. MAP Message Flow

The diagram below depicts a typical MAP message flow for data transfer between the "Network Node" and the Restcomm platform to implement a menu driven application. If you would like to read more about mobile-intiated (and network-initated) GMLC operations and the use of MAP GMLC services, please refer to [3GPPTS 24.090] in the references section.

GMLC service begins when the network send an HTTP (GET/POST) request tp GMLC Gateway. The message flow involves the following steps:

1. The Network sends a 'TCAP Begin' message with the Component 'ANY_TIME_INTERROGATION_REQUEST' to the Restcomm platform. The Restcomm platform invokes GMLC application logic.
2. The HLR receives the mobile msisdn.
3. Based on the msisdn, the HLR performs the predefined logic and sends a response back to the GMLC Gateway by 'ANY_TIME_INTERROGATION_REQSPONSE' and terminates the 'TCAP dialogue'.

2.4. GMLC Gateway

Existing MSC, VLR, and HLR network elements are proprietary and run on non-standard operating environments located in trusted operator's zones that make it difficult to build and deploy new applications. Also, these network elements do not provide the tools and interfaces needed to access and retrieve data from content providers over the Internet. The GMLC Gateway connects to the MSC, VLR, or HLR and enables the flow of GMLC messages to be extended to an open, standards-based application server located in the IP network. The AS also provides the tools and interfaces to enable access to the content providers through the Internet.

2.5. Restcomm GMLC

2.5.1. Major Features

Restcomm 's implementation of GMLC Gateway is the first and only open source GMLC Gateway with a host of rich features and advantages.

Java-based

Restcomm GMLC is the only Java based GMLC Gateway. It is robust and reliable and can be installed on any Operating System that supports Java (JDK 7 and SCTP).

Open Source

The Software is open-source, giving you the freedom to understand the code and customise it to your enterprise needs. It is supported by a vibrant Open source community.

Carrier Grade Performance

Restcomm GMLC has been deployed at telecom operators around the world and is processing billions of GMLC transactions each day. A single RestComm GMLC node can process 1500's of GMLC/sec and can be adapted to the needs of telecom service providers of different sizes in any country reducing your CAPEX and OPEX costs.

Cloud Ready

Restcomm GMLC is Cloud-ready. It can be deployed on dedicated hardware, private cloud infrastructure or public IaaS such as AWS.

Network Push

Restcomm GMLC supports network/application/service initiated GMLC request. The request can be just HTTP PUSH.

SS7 Hardware Cards

Restcomm GMLC can be used with Intel family boards (Dialogic SS7 cards) or Zaptel/Dahdi compatible TDM devices (Digium, Sangoma). For production its recommended to use Dialogic boards only.

SIGTRAN (M3UA)

It also has in-built support for SIGTRAN (M3UA using SCTP).

HTTP interface

HTTP interface is a common interface that can be used for connection with service applications.

Easy Configuration and Management

Restcomm GMLC comes with an efficient Command Line Interface (CLI) tool allowing you to completely configure the Gateway at run-time and manage it using simple commands rather than do everything manually. Restcomm GMLC also comes with a Graphical User Interface that will allow you to configure, monitor and manage the Gateway through a convenient user-friendly interface.

2.5.2. Technical Specifications

Restcomm GMLC is not restricted by Transaction Per Second model. The only restricting factor is memory + CPU capacity of the host servers, third-party applications or the underlying database service.

- Restcomm GMLC supports as many as 1073741823 incoming and 1073741823 outgoing concurrent sessions/dialogs.
- Restcomm GMLC supports unlimited E1 links and the only limiting factor is the underlying TDM board used.
- Restcomm GMLC SCTP supports as many associations as supported by the underlying Operating System. Can be setup in multihome.
- Restcomm GMLC M3UA can be confgured to have as many ASP's / IPSP's as needed by the system.
- Restcomm GMLC SCCP can be confgured to have virtually unlimited Global Title Translation rules and also supports wild characters for partial matching of Global Title digits.

2.5.3. HTTP Transfer Mechanism

The Restcomm GMLC Gateway makes use of HTTP protocol between the gateway and the third-party applications (or Value Added Service Modules). Restcomm GMLC Gateway receives the GMLC request from the third-party applications and then translates these requests to SS7 MAP and route to a corresponding HLR. The HTTP callback mechanism allows the third-party Application to be agnostic to Operating System, Programming Language and Framework. The third-party Application can be either of the following technologies on any Operating System:

- Apache Tomcat, JBoss AS, Oracle Application Server, IBM Websphere etc for JSP/Servlet on Java
- PHP
- Microsoft IIS for ASP

Chapter 3. Running

3.1. Running the Gateway

Procedure: Run Restcomm GMLC

1. Pre-requisite:
    - You must have Restcomm GMLC installed as explained in the Installation Guide.
    - If you are using the SS7 board on server, you must ensure that the `java.library.path` variable is set to point to the directory containing the native component. Alternatively you can copy it to the JBoss native library path manually.
2. All you have to do to start the Gateway is start the JBoss Application Server. To start the JBoss Server you must execute the *run.sh* (Unix) or *run.bat* (Microsoft Windows) startup script in the installation directory *restcomm-gmlc-/jboss-5.1.0.GA/bin*. Note that this will start the server in the default profile. The "default" profile is a clean profile where you start from scratch and configure the entire SS7 Stack and GMLC Gateway to suit your requirements.
3. Result: If the service started properly you should see the following last few output lines in the Unix terminal or Command Prompt depending on your environment:

```
22:23:11,583 INFO  [DeploymentMBeanImpl] (main) Installed
DeployableUnitID[url=file:/home/vinu/restcomm-gmlc-<version>/jboss-
5.1.0.GA/server/default/deploy/restcomm-gmlc-gateway/services-DU-6.1.5.GA.jar/]
22:23:11,874 INFO  [ServiceManagementImpl] (main) Activated
ServiceID[name=restcomm-gmlc-cdr,vendor=org.mobicents,version=1.0]
22:23:11,976 ERROR [STDERR] (pool-28-thread-1) QUERRY: 000000 CREATE TABLE
GMLC_GW_CDRS (ID VARCHAR(150) NOT NULL, L_SPC INT, L_SSN SMALLINT, L_RI SMALLINT,
L_GT_I SMALLINT, L_GT_DIGITS VARCHAR(18), R_SPC INT, R_SSN SMALLINT, R_RI SMALLINT,
R_GT_I SMALLINT, R_GT_DIGITS VARCHAR(18), SERVICE_CODE VARCHAR(10), OR_NATURE
SMALLINT, OR_PLAN SMALLINT, OR_DIGITS VARCHAR(18), DE_NATURE SMALLINT, DE_PLAN
SMALLINT, DE_DIGITS VARCHAR(18), ISDN_NATURE SMALLINT, ISDN_PLAN SMALLINT,
ISDN_DIGITS VARCHAR(18), VLR_NATURE SMALLINT, VLR_PLAN SMALLINT, VLR_DIGITS
VARCHAR(18), IMSI VARCHAR(100), TERMINATE_REASON VARCHAR(60), TSTAMP TIMESTAMP  NOT
NULL , DIALOG_ID BIGINT, PRIMARY KEY(ID,TSTAMP));
22:23:12,135 INFO  [ServiceManagementImpl] (main) Activated
ServiceID[name=restcomm-gmlc,vendor=org.mobicents,version=1.0]
22:23:12,395 INFO  [GMLCPropertiesManagement] (main) Loading GMLC Properties from
/home/vinu/restcomm-gmlc-6.1.5.GA/jboss-
5.1.0.GA/server/default/data/GMLCManagement_GMLCproperties.xml
22:23:12,395 WARN  [GMLCPropertiesManagement] (main) Failed to load the GMLC
configuration file.
/home/vinu/restcomm-gmlc-6.1.5.GA/jboss-
5.1.0.GA/server/default/data/GMLCManagement_GMLCproperties.xml (No such file or
directory)
22:23:12,396 INFO  [ShortCodeRoutingRuleManagement] (main) Loading short code
routig rule configuration from /home/vinu/restcomm-gmlc-6.1.5.GA/jboss-
5.1.0.GA/server/default/data/GMLCManagement_scroutingrule.xml
22:23:12,397 WARN  [ShortCodeRoutingRuleManagement] (main) Failed to load the short
code routig rule configuration file.
/home/vinu/restcomm-gmlc-6.1.5.GA/jboss-
5.1.0.GA/server/default/data/GMLCManagement_scroutingrule.xml (No such file or
directory)
22:23:12,400 INFO  [GMLCManagement] (main) Started GMLCManagement
22:23:12,419 INFO  [ShellServer] (main) Starting SS7 management shell environment
22:23:12,430 INFO  [ShellServer] (main) ShellExecutor listening at /127.0.0.1:3435
22:23:12,498 INFO  [Http11Protocol] (main) Starting Coyote HTTP/1.1 on http-
127.0.0.1-8080
22:23:12,529 INFO  [AjpProtocol] (main) Starting Coyote AJP/1.3 on ajp-127.0.0.1-
8009
22:23:12,541 INFO  [ServerImpl] (main) JBoss (Microcontainer) [5.1.0.GA (build:
SVNTag=JBoss_5_1_0_GA date=200905221634)] Started in 1m:11s:118ms
```

4. If you are starting GMLC-1.0.0-SNAPSHOT for the first time, SS7 is not configured. You can use either the Shell Client or the GUI to connect to GMLC-1.0.0-SNAPSHOT and configure the SS7 Stack, GMLC parameters and Routing Rules. Once configured, the state and configuration of SS7 and GMLC are both persisted which stands a server re-start operation. The next chapter will discuss in detail about configuring SS7 and the GMLC Gateway.

Procedure: Stop the Gateway

1. To stop the Restcomm GMLC , you must shut down the JBoss Application Server. To shut down the server(s) you must execute the `shutdown.sh -s` (Unix) or `shutdown.bat -s` (Microsoft Windows) script in the installation directory *restcomm-gmlc-/jboss-5.1.0.GA/bin*.
2. If the server stopped properly, you will see the following three lines as the last output in the Unix terminal or Command Prompt:

```
[Server] Shutdown complete
Halting VM
```

3.2. Running the Gateway - Simulator Profile

The Restcomm GMLC offers you an option to run the Gateway with a "simulator" profile for testing purpose. The "simulator" profile is a pre-configured profile to work with the jss7-simulator. Starting the Gateway with the "simulator" profile is similar to the steps explained for the "default" profile except that you must pass the string value "simulator" to the -c command line option when invoking the run script.

```
[bin]$ ./run.sh -c simulator
```

By default, the GMLC Simulator profile is configured for use in Linux systems. For using it in Microsoft Windows systems, you must configure the parameters as explained below.

Open the file *restcomm-gmlc-<version>/jboss-5.1.0.GA/server/simulator/data/SCTPManagement_sctp.xml* and replace in two places, the parameter `ipChannelType="0"` with `ipChannelType="1"` to enable TCP connection instead of SCTP since Windows does not support SCTP. If you are using in a Linux system, there is no modification required to the settings.

3.3. Running GMLC Examples in Simulator

If you are not familiar with the RestComm jss7 Simulator, you can find instructions about using the jss7-simulator in the Restcomm jSS7 User Guide. You will also find example test cases explained in detail in the jSS7 User Guide. In this section you will find a sample GMLC Pull and GMLC Push examples explained using the jSS7 Simulator.

Procedure: Running RestComm jSS7 Simulator - GMLC Pull Example

1. Change the working directory to the bin folder in the Simulator's installation directory.

```
[vinu@vinu-neha ~]$ cd restcomm-gmlc-<version>/tools/restcomm-ss7-simulator/bin
```

2. Ensure that the *run.sh* start script is executable.

```
bin$ chmod +x run.sh
```

3. Execute the *run.sh* Bourne shell script with the command `./run.sh gui`.

```
bin$ ./run.sh gui
```

   This will launch the Simulator GUI Application.

4. When the GUI shows up, select "main" (default) as host name [or type "win" as host name under Windows] and press the 'Start' button. The Simulator is already pre-configured to connect to the GMLC Gateway (running in simulator profile). Press 'Run test' and again click on 'Start' in the next screen. The Simulator will connect to GMLC (via m3ua protocol). The Low level part is configured to SCTP (not TCP) protocol and hence you can test the GMLC in a Linux environment. To test under Windows OS, you must change the SS7 simulator settings to TCP.
5. After approximately 30 seconds you will see the state of the Simulator change to "M3UA connection is active" as in figure below:

   [GMLC SS7 Simulator ACTIVE] | *images/GMLC_SS7_Simulator_ACTIVE.png*

   Figure 1. GMLC SS7 Simulator - Active

6. Restcomm GMLC is configured with a routing rule for *519#. Dial *519# in your Simulator GUI and press 'Send ProcessUnstructuredRequest'. The example will respond to you with the message "Hello World 1. Balance 2. Texts Remaining".

   [GMLC SS7 Simulator Process Unst request] |

images/GMLC_SS7_Simulator_Process_Unst_request.png

Figure 2. GMLC SS7 Simulator - Process Unstructured Request

7. Now Dial 1 in your Simulator GUI and press 'Send UnstructuredResponse'. You should get a response "Thank you!".

[GMLC SS7 Simulator Unstruc request] | *images/GMLC_SS7_Simulator_Unstruc_request.png*

Figure 3. GMLC SS7 Simulator - Unstructured Request

Procedure: Running Restcomm GMLC Simulator (HLR) - GMLC Push Example

1. You must first start the Restcomm GMLC in simulator profile.

```
[vinu@vinu-neha ~]$ cd restcomm-gmlc-<version>/jboss-5.1.0.GA/bin
[vinu@vinu-neha bin]$./run.sh -b 127.0.0.1 -c simulator
```

1. To send a PUSH request go to http://localhost:8080/jmx-console/ and click the link `org.mobicents.gmlc.example` in the left menu. Then open the MBean `'name=HttpPush'`.
2. MBean provides two operations: 1) `sendNotify` to push Notification and 2) `sendRequest` to push GMLC menu based tree. The parameter `Isdn` is the MSISDN to which Notify or Request is to be sent.

[Restcomm GMLC simulator Notify] | *images/Restcomm-GMLC-simulator_Notify.png*

Figure 4. RestComm GMLC Simulator - Notify

You can simulate a simple Notify dialog by following the below steps:

- Fill the ISDN field with a preferred ISDN number, for example "1111" is good for SS7 Simulator. Now press "Apply changes".
- Perform "reset" operation. Perform "sendNotify" operation with parameters: String=<Text of your notification>, boolean=false, int=60000 and String=<any random string>. Parameters definition is as below
  - 1st String is GMLC message that you want to push to mobile
  - 2nd Boolean if set to true means GMLC Gw will send empty TCAP Begin and try to establish dialog before sending actual message.
  - 3rd Int is custom invoke timeout. User must respond within this period else GMLC Gw will terminate Dialog and Application will get appropriate error message
  - 4th String is random string that is stored at GMLC Gw side as custom object.
  - When ever response comes back, GMLC Gw will include this custom string in XML Payload.
  - Perform "close" operation.

You will now find a notification at the SS7 Simulator.

You can also simulate more complicated scenarios like pushing the tree based menu to user and expecting some input from users by calling `sendRequest`. The below Class provides more explanation

for attributes and operations of HttpPush.

3.4. Running the Shell

You must start the Shell client and connect to the managed instance prior to executing commands to configure the Gateway. Shell can be started by issuing the following command from *restcomm-gmlc-/jboss-5.1.0.GA/bin* directory:

```
[$] ./ss7-cli.sh
```

Once console starts, it will print following information and await further commands:

```
version=2.0.0-SNAPSHOT,name=mobicents CLI,prefix=restcomm,vendor=TeleStax
mobicents>
```

Before issuing further commands you must connect to a managed instance. For more details on connecting to an instance and for a list of all supported commands and details on configuring the SS7 stack refer to the RestComm SS7 Stack User Guide.

3.5. Connect to a new Instance

You can connect to a new instance by entering the IP:Port values and the login credentials in the top left corner of the GUI. However please note that this feature is not available in this release but will be fully functional in the next release.

3.6. Authentication

Restcomm GMLC GUI Management Security is based on the JBoss Security Framework. However please note that the feature is not fully functional yet and you will not be able to sign-out or sign-in using the login panel at the top right corner of the GUI. Future releases will offer a full implementation.

As of now, there is basic authentication offered (which is based on the JBoss Security framework). When you try to start the Web Console, you will be prompted to enter login credentials. These credentials can be configured in the files *jmx-console-roles.properties* and *jmx-console-users.properties* located at *restcomm-gmlc-<version>/jboss-5.1.0.GA/server/<profile>/conf/props/*.

You can also change the authentication from flat file system to database by making necessary configurations in the file *restcomm-gmlc-<version>/jboss-5.1.0.GA/server/<profile>/conf/login-config.xml*.

For detailed instructions and to know more about JBoss Security Framework please refer to the JBoss Installation Guide here.

Deafult user-id and password for GUI Management Console is admin and admin. You can change the user-id and password in files *jmx-console-roles.properties* and *jmx-console-users.properties* located at *restcomm-gmlc-<version>/jboss-5.1.0.GA/server/<profile>/conf/props/*

Chapter 4. Configuring

You must fine-tune Memory and Database settings for better performance before using Restcomm GMLC in production. Once you complete setting up the Gateway you must configure the SS7 Stack and GMLC paramters. Restcomm GMLC comes with a convenient user-friendly Graphical User Interface (GUI) and a Command Line Interface (CLI) that will allow you to configure, monitor and manage the Gateway. While the CLI tool allows complete configuration and control of the Gateway, the GUI-based management enhances the usability of the Gateway and gives you the ability to configure and manage the GMLC Gateway dynamically. This chapter will explain how to manage the Gateway effectively using both the GUI and the CLI.

4.1. Memory Settings

You should fine tune the JVM memory settings based on your needs but we recommend you allocate a minimum of 3 GB for initial and maximum heap size. These settings are specified in the file[path]*restcomm-gmlc-/jboss-5.1.0.GA/bin/run.conf*.

-Xms3072m

Initial heap size, set in megabytes

-Xmx3072m

Maximum heap size, set in megabytes

4.2. JSupported Java Version

GMLC Gateway can run only with Java 7 JRE or JDK. We refered Oracle Java 7 JDK.

4.3. Configuring JSLEE http-client RA

Restcomm GMLC acts as a HTTP Client to achieve GMLC pull by sending a HTTP POST/GET request to GMLC gateway. You must configure the HTTP Client JSLEE Resource Adaptor's properties to suit your requirements. Please refer to the SLEE RA HTTP Client User Guide available in *restcomm-gmlc-/docs/slee/RestComm_SLEE_RA_HTTP_Client_User_Guide.pdf*.

4.4. Configuring the SS7 Stack

You must configure the SS7 Stack prior to configuring GMLC. For details on configuring the SS7 Stack please refer to the RestComm SS7 Stack User Guide. The RestComm SS7 Stack User Guide lists all available Shell commands and GUI operations to configure SS7. In addition, help files are also available for every Shell command providing all details relevant to the command.

Appendix A: Revision History