User Guide to Restcomm SMSC

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 SMSC` `, or contact the authors.

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

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

Restcomm SMSC is an Open Source Java based SMSC Gateway built on a modern extensible middleware platform and provides easy integration with any third party communication protocol and platform. Restcomm SMSC enables mobile operators to provide core SMSC features like subscriber to subscriber SMS messaging, broadcasting campaigns and group messaging [1: Group Messaging can be achieved using RestComm. For more details please contact info@telestax.com.] between subscribers. In addition it also enables value added services through integration with OTT social networks and microblogs such as Facebook and Twitter.

Restcomm SMSC is an easy-to-install and easy-to-deploy platform that can be set up and configured very quickly. Since it is built on a modern and extensible JSLEE platform it automatically provides out-of-the-box integration with various communication protocols such as SIP, Diameter, HTTP, XCAP, XMPP, MGCP and others in addition to the base SMPP protocol for connectivity to ESMEs.

Restcomm SMSC stores messages using Cassandra database. The database is used for storing unsent messages, messages successfully sent and messages that failed to be sent.

Restcomm SMSC is cloud-ready. It can be deployed on dedicated hardware, private cloud infrastructure or public laaS such as AWS. Restcomm SMSC supports TDM hardware offered by major vendors in the market, namely Intel family boards (Dialogic) and Zaptel/Dahdi (Digium, Sangoma).

Restcomm SMSC is based on the robust and proven Restcomm JAIN SLEE 1.1 Server and Restcomm jSS7 Stack. Restcomm JAIN SLEE Server is a highly scalable event-driven application server with a robust component model and fault tolerant execution environment. It provides a set of connectors to a variety of networks elements: SS7 MAP, TCAP, INAP, ISUP, SMPP, XMPP, SIP, MGCP, HTTP, XDM, XCAP, Diameter and many others. It is fully compliant with JSR 240 (JSLEE 1.1). Restcomm jSS7 is a software based implementation of the SS7 protocol. It provides implementation for Level 2 and above in the SS7 protocol Stack. Restcomm jSS7 Stack User Guide is bundled within and you can refer to the guide for more details on the Stack.

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. SMSC Gateway

SMSC (Short Message Service Center) is a telecommunications network facility responsible for handling the SMS operations of a wireless network. When a subscriber sends an SMS message from a mobile phone, the message first reaches the SMS Center which then forwards the message to the desired destination. The SMS Center is responsible for routing the SMS messages and regulating the process. If the recipient is not reachable (the phone is switched off or out of coverage area etc.) the SMS Center will store the message to a database and forward it to the recipient when available. Messages that are delivered successfully and messages that failed to be delivered are stored in an archive database for logging purposes.

2.2. Restcomm SMSC

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

Java-based

Restcomm SMSC is the only Java based SMSC 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.

SS7 Hardware Cards

Restcomm SMSC supports E1/T1 links via SS7 cards (Dialogic SS7 cards) or Zaptel/Dahdi compatible TDM devices (Digium, Sangoma).

SIGTRAN (M3UA)

It also supports connectivity to MSC/HLR via SIGTRAN links (M3UA using SCTP).

Charging enabled (Diameter)

Pre paid based charging is enabled (optional) via Diameter Ro and Base protocol. RestComm SMSC easily integrates with RestComm Convergent Charging Server over Diameter

Cloud-ready

Restcomm SMSC is cloud-ready and can also be deployed on a private cloud infrastructure.

Easy Configuration and Management

Restcomm SMSC comes with an efficient Command Line Interface (CLI) tool allowing you to easily configure the gateway at run-time and manage it using simple commands rather than do everything manually. Restcomm SMSC also includes an efficient Web Interface administration tool for easy management and monitoring through a Web console.

Major Features

Restcomm SMSC enables mobile operators to provide core SMSC features like subscriber to subscriber SMS messaging, broadcasting campaigns and group messaging between subscribers.

- Restcomm SMSC can handle mobile originated SMS and deliver them to the intended recipient (mobile, ESME or SIP). It also provides a flexible routing mechanism to route messages to/from ESME over SMPP protocol. Restcomm SMSC supports normal style (Server side) SMPP connection. It can also act as ESME initiating connection to third party SMSC (Client side). Submit_Sm, Data_Sm, Submit_Multi and Deliver_Sm messages are supported.
- Restcomm SMSC supports SIP connection through JSLEE SIP RA.
- For incoming from SMPP ESME messages, Restcomm SMSC supports StoreAndForward, Datagramm and Transactional modes and a special ForwardAndStore mode for reducing the database load.

  When a message comes from SMPP, the SMSC checks the `EsmClass` field in the SMPP message. For Datagramm and Transactional modes, the SMSC will try to immediately deliver the message to a destination. If it is in Transactional mode, the SMSC will also return a delivery result (success/failure) to the ESME. Such messages are not stored in the Cassandra database. In case of delivery failure the message will be dropped and no more delivery attempts will be made.

  If the `EsmClass` field indicates StoreAndForward (or default) mode, more than one attempt will be made to deliver the message (if the first attempt fails). Such messages will be stored in the Cassandra database. The SMSC can be configured in two modes for processing of StoreAndForward messages as shown below:

  StoreAndForward (normal) mode

  In the StoreAndForward mode, an incoming "StoreAndForward" message will always be stored in a database and scheduled for delivery after few seconds. This mode is fully compliant with SMPP specification.

  ForwardAndStore (fast) mode

  In the ForwardAndStore mode, an incoming "StoreAndForward" message will be attempted for delivery immediately, without storing in the database and therefore without any delay. If this first delivery attempt fails, the message will be stored in a database for futher scheduling.

- Messages that come from SS7 or SIP are processed the same way as "StoreAndForward" messages that come from SMPP ESME.
- Restcomm SMSC supports Non Transparent Home Routing as explained in 3GPP Specification version 23.840 5.2.3.
- SMSC has support for SSL for SMPP connection (both client and server side).
- Restcomm SMSC can be configured for congestion control to reject messages when the number of messages in processing exceeds a configurable limit (`MaxActivityCount`).

  When the SMSC is configured in ForwardAndStore (fast) mode, SMPP originated messages will be first rejected. Then when the number of messages in processing further increases, SIP

originated messages will be rejected and finally SS7 originated messages will be rejected.

When SMSC is configured in StoreAndForward (normal) mode, only datagramm and transactional SMPP originated messages will be rejected. However StoreAndForward messages and all SS7/SIP originated messages will be stored in the Cassandra database and the scheduling for delivery of these messages will be stopped until the number of messages in processing does not exceed the configured limit.

- Restcomm SMSC has a congestion control at ESME side. Each ESME can be configured for rate limit per a second, a minute, a hour or a day.
- Restcomm SMSC supports ReportSMDeliveryStatus and AlertServiceCentre requests after mobile delivery failure.
- Restcomm SMSC supports GSM7, GSM8 and UCS2 message encoding.
- Restcomm SMSC itself splits ESME and SIP originated long length messages before delivering to mobile phones, if a message does not contain User Data headers (UDH).
- Multi-tenancy support: Restcomm SMSC supports a model of message network as a splitted to several logical subnetworks and SMSC messages will be logically processed only in one of this logical subnetwork. SMPP ESMEs, SCCP SAPs (Service Access Points) and SIP connector will belong to one of such subnetworks (A subnetwork is identified by a digital value - networkId). Any message that has come to SMSC GW from some port (ESME, SS7, SIP) will belong to networkId (subnetwork) from networkId of the port the message has come to SMSC. Then the message will be routed into the port (ESME, SS7, SIP) which belongs to the same networkId (subnetwork).
- Additionally, when enabled with proper administrator privileges, Restcomm SMSC allows convenient group messaging between subscribers (e.g. family, friends, colleagues) .
- Restcomm SMSC is built on the RestComm jSS7 stack which provides advanced SCCP routing rules to map short codes to MSISDN, allowing users to send an SMS to a short-code instead of a complete MSISDN.
- It also enables value added services through integration with OTT social networks and microblogs such as Facebook and Twitter.
- A single Restcomm SMSC node can process up to 1000 SMS/sec. Multiple SMSC nodes can be arranged in a cluster across one or more geographically distributed data centers to scale up throughput and provide various levels of redundancy, high availability and fault tolerance.

Restcomm SMSC can be adapted to the needs of telecom service providers of different sizes in any country.

2.3. Session Initiation Protocol (SIP) Support

IP Short Message Gateway (IP-SM-GW), which is in line with relevant 3GPP 23.824 specifications, acts as a bridge managing the origination and termination of SMS messages between circuit-switched and IP-based networks (over SIP). With this solution, mobile operators can make use of existing messaging platforms such as short message service centers (SMSCs) to deliver IP-based messaging. The IP-SM-GW is also essential for rolling out rich communication services (RCS), which enable operators to deliver sticky messaging services like presence-enabled address book, SIP-

based chat services and mobile instant messaging.

RestComm SMSC has support for SIP messagaing and can send short messages from SIP to SS7 or ESME and vice-a-versa. RestComm SMSC can easily be integarted with RestComm RestComm via SIP to enable developers to rapidly build voice, video, WebRTC, USSD, SMS, fax and rich messaging applications. For more details about RestComm you may refer to, TeleStax website.

Chapter 3. Architecture

Restcomm SMSC is based on robust and proven Restcomm JAIN SLEE 1.1 Server and Restcomm jSS7 Stack and provides easy integration with any third party communication protocol and platform.

Restcomm JAIN SLEE Server is a highly scalable event-driven application server with a robust component model and fault tolerant execution environment. It provides a set of connectors to a variety of networks elements: SS7 MAP, TCAP, INAP, ISUP, SMPP, XMPP, SIP, MGCP, HTTP, XDM, XCAP, Diameter and many others. It is fully compliant with JSR 240 (JSLEE 1.1).

Restcomm jSS7 is a software based implementation of the SS7 protocol. It provides implementation for Level 2 and above in the SS7 protocol Stack. Restcomm jSS7 Stack User Guide is bundled within and you can refer to the guide for more details on the Stack.

The diagram below depicts a high level design of Restcomm SMSC .

Restcomm SMSC provides six basic services of an SMSC as shown in the figure above. Each of these six services are implemented in the Gateway as a module that can be activated or deactivated based on the needs of an operator. All these services interact with the Cassandra database for storing data (messages not yet sent and an archive for sent/failed to send messages) and retreiving data for messages that are pending to be sent.

Restcomm SMSC can accept SMPP (incoming BIND) from ESMEs or can also initiate SMPP (outgoing BIND) to third party SMSC and hence itself acting as an ESME. Restcomm SMSC provides intelligent routing rules that can route SMS between various SMPP connections or between SMPP, GSM and SIP Network.

Service Modules

Mo (Mobile originated) module

Handles mobile originated SMS.

Mt (Mobile terminated)

Delivers SMS to mobile.

Rx SMPP module

Listens for incoming SMS from ESME/third party SMSC and routes them to Mt module (GSM), other ESME/third party SMSC or SIP Client depending on the routing rule.

Tx SMPP module

Delivers SMS to external ESME/third party SMSC received from Mo Module (GSM), ESME/third Party SMSC or SIP Client.

Rx SIP module

Listens for incoming SMS from SIP Client and routes them to Mt module (GSM), other ESME/third party SMSC or SIP Client depending on the routing rule.

Tx SIP module

Delivers SMS to SIP Client received from Mo Module (GSM), ESME/third Party SMSC or other SIP Client. | |

Administration

CLI

Restcomm SMSC comes with a Command Line Interface (CLI) that provides easy to use commands to manage and monitor the SS7 Stack and the SMSC Gateway.

Web Interface

Restcomm SMSC also includes an easy to use Web Interface administration tool that allows you to manage and monitor the Gateway via a convenient Web console.

Campaign Tool

In addition, future releases will also offer a Web based campaign tool to define and manage campaigns. |

3.1. Database Table Structure

Restcomm SMSC stores messages using Cassandra database. The database is used for storing unsent messages, messages successfully sent and messages that failed to be sent.

The Restcomm SMSC creates three tables for every new day with the date suffixed to the names of the tables as `DST_SLOT_TABLE_yyyy_mm_dd`, `SLOT_MESSAGES_TABLE_yyyy_mm_dd` and `MESSAGES_yyyy_mm_dd` where `yyyy_mm_dd` is the date in that format.

Apart from the above tables created everyday, the database also comprises of three other tables namely `CURRENT_SLOT_TABLE`, `SMPP_SMS_ROUTING_RULE` and `SIP_SMS_ROUTING_RULE` created when the SMSC is started for the first time. The former is used for storing system-wide data and the latter two for storing the routing rules.

3.1.1. CURRENT_SLOT_TABLE

Table 1. `CURRENT_SLOT_TABLE`

Column Name	Data Type	Description
ID	INT	Acts as a primary key and is unique for each entry. The value acts as an identifier of content: 0 indicates the current due slot that the SMSC processes now and 1 indicates the last assigned messageId.
NEXT_SLOT	BIGINT	The next slot value.

3.1.2. SMPP_SMS_ROUTING_RULE

Table 2. `SMPP_SMS_ROUTING_RULE`

Column Name	Data Type	Description
ADDRESS	Text	Acts as a primary key and is unique for each entry. Stores the address for which the messages are being routed.
CLUSTER_NAME	Text	The name of the SMPP Cluster.

3.1.3. SIP_SMS_ROUTING_RULE

Table 3. `SIP_SMS_ROUTING_RULE`

Column Name	Data Type	Description
ADDRESS	Text	Acts as a primary key and is unique for each entry. Stores the address for which the messages are being routed.
CLUSTER_NAME	Text	The name of the SIP Cluster. This is not used as of now and all SMS message are routed through a single SIP stack.

3.1.4. DST_SLOT_TABLE_yyyy_mm_dd

This table contains a list of `DUE_SLOT` values for `TARGET_ID` to store the data for this day.

Table 4. `DST_SLOT_TABLE_yyyy_mm_dd`

Column Name	Data Type	Description
TARGET_ID	ASCII	Acts as a primary key and is made up of ADDR_DST_DIGITS+""*ADDR_DST_TON*""+ADDR_DST_NPI.
DUE_SLOT	BIGINT	All new incoming messages will be added into this DUE_SLOT if it is not yet processed. If DUE_SLOT has been processed already or is absent, a new DUE_SLOT will be assigned.

3.1.5. SLOT_MESSAGES_TABLE_yyyy_mm_dd

This table stores the messages that are scheduled for delivery. The messages are not deleted after delivery.

The fields "DUE_SLOT", "TARGET_ID", "ID" together act as the primary key.

Table 5. `SLOT_MESSAGES_TABLE_yyyy_mm_dd`

Column Name	Data Type	Description
ID	UUID	Record Identifier.
TARGET_ID	ASCII	Made up of ADDR_DST_DIGITS+””*ADDR_DST_TON*””+ADDR_DST_NPI.
DUE_SLOT	BIGINT	For which the messages will be loaded for delivering.
IN_SYSTEM	INT	0 - idle state, 1 - delivery in progress, 2 - delivery finished (by success or failure)
SMSC_UUID	UUID	Id of the SMSC session (from start to stop), this is needed to know which session has launched the delivery of a message.
ADDR_DST_DIGITS	ASCII	Destination address digits.
ADDR_DST_TON	INT	SMPP style TON (type of number) of destination address.
ADDR_DST_NPI	INT	SMPP style Numbering Plan Indicator of destination address.
ADDR_SRC_DIGITS	ASCII	Originating address digits.
ADDR_SRC_TON	INT	SMPP style TON (type of number) of source address.
ADDR_SRC_NPI	INT	SMPP style Numbering Plan Indicator of source address.
DUE_DELAY	INT	Duration (in seconds) after which a new delivery attempt will be done. If the SMS has just arrived in the system, this value is 0.
ALERTING_SUPPORTED	BOOLEAN	The value is 'true' if SMSC was successfully registered at HLR after delivery failure. However this field is currently not being used because this demands extra database access.
MESSAGE_ID	BIGINT	A unique message ID assigned by SMSC (since the SMSC started).
MO_MESSAGE_REF	INT	SMS TPDU Message Reference field.

Column Name	Data Type	Description
ORIG_ESME_ID	TEXT	SMSC internal name of origination ESME (empty for MO messages).
ORIG_SYSTEM_ID	TEXT	SMPP name of origination ESME (empty for MO messages).
DEST_CLUSTER_NAME	TEXT	Name of cluster for destination ESME terminated massages (empty for MT messages).
DEST_ESME_ID	TEXT	SMSC internal name of destination ESME (empty for MT messages).
DEST_SYSTEM_ID	TEXT	SMPP name of destination ESME (empty for MT messages).
SUBMIT_DATE	TIMESTAMP	Time when a message was received by SMSC.
DELIVER_DATE	TIMESTAMP	Time when a message was sent from SMSC (null if message failed to deliver).
SERVICE_TYPE	TEXT	SMPP parameter (service_type) for ESME originated messages.
ESM_CLASS	INT	Indicates Message Mode (Messaging Mode==Datagram, Forward or Store and Forward mode) and Message Type (MessageType==some flags including UDH indicator).
PROTOCOL_ID	INT	Protocol Identifier SMPP parameter (TP-Protocol-Identifier files for GSM).
PRIORITY	INT	SMPP parameter (priority_flag).
REGISTERED_DELIVERY	INT	SMPP parameter (registered_delivery).
REPLACE	INT	SMPP parameter (replace_if_present_flag).
DATA_CODING	INT	data_coding scheme.
DEFAULT_MSG_ID	INT	SMPP parameter (sm_default_msg_id).
MESSAGE	BLOB	Message text in source style that has been received from EMSE or from MS.

Column Name	Data Type	Description
OPTIONAL_PARAMETERS	TEXT	TLVs.
SCHEDULE_DELIVERY_TIME	TIMESTAMP	SMPP parameter (schedule_delivery_time) - time when SMSC should start a delivery (may be null if immediate message delivery).
VALIDITY_PERIOD	TIMESTAMP	The validity period of this message. If ESME has not defined (or for MO messages) this field is filled by default SMSC settings.
IMSI	ASCII	From SRI response.
CORR_ID	ASCII	This field is used for keeping of correlationId value for home routing mode. This table will keep this value when StoreAndForward mode for time between storing of message and scheduling it for delivering.
NNN_DIGITS	ASCII	NetworkNodeNumber = MSC that serves MS – from SRI response.
NNN_AN	INT	
NNN_NP	INT	
SM_STATUS	INT	Error Code value for the last attempt (0==no attempts yet). For more details on Error Codes please refer to Appendix A, Error Codes
SM_TYPE	INT	0-ESME terminated, 1-MT
DELIVERY_COUNT	INT	Delivery attempt count. (this will be==1 if a message was delivered in one go)

3.1.6. MESSAGES_yyyy_mm_dd

This table archives the messages that have been delivered successfully or whose delivery failed.

The fields "ADDR_DST_DIGITS", "ID" together act as the primary key.

Table 6. `MESSAGES_yyyy_mm_dd`

Column Name	Data Type	Description
ID	UUID	Record Identifier.
TARGET_ID	ASCII	Made up of ADDR_DST_DIGITS+””*ADDR_DST_TON*””+ADDR_DST_NPI.
DUE_SLOT	BIGINT	For which the messages will be loaded for delivering.
IN_SYSTEM	INT	Not used in this table.
SMSC_UUID	UUID	Id of the SMSC session (from start to stop), this is needed to know which session has launched the delivery of a message.
ADDR_DST_DIGITS	ASCII	Destination address digits.
ADDR_DST_TON	INT	SMPP style TON (type of number) of destination address.
ADDR_DST_NPI	INT	SMPP style Numbering Plan Indicator of destination address.
ADDR_SRC_DIGITS	ASCII	Originating address digits.
ADDR_SRC_TON	INT	SMPP style TON (type of number) of source address.
ADDR_SRC_NPI	INT	SMPP style Numbering Plan Indicator of source address.
DUE_DELAY	INT	Duration (in seconds) after which a new delivery attempt will be done - value before the last delivery attempt.
ALERTING_SUPPORTED	BOOLEAN	The value is 'true' if SMSC was successfully registered at HLR after delivery failure. However this field is currently not being used because this demands extra database access.
MESSAGE_ID	BIGINT	A unique message ID assigned by SMSC (since the SMSC started).
MO_MESSAGE_REF	INT	SMS TPDU Message Reference field.
ORIG_ESME_ID	TEXT	SMSC internal name of origination ESME (empty for MO messages).

Column Name	Data Type	Description
ORIG_SYSTEM_ID	TEXT	SMPP name of origination ESME (empty for MO messages).
DEST_CLUSTER_NAME	TEXT	Name of cluster for destination ESME terminated massages (empty for MT messages).
DEST_ESME_ID	TEXT	SMSC internal name of destination ESME (empty for MT messages).
DEST_SYSTEM_ID	TEXT	SMPP name of destination ESME (empty for MT messages).
SUBMIT_DATE	TIMESTAMP	Time when a message was received by SMSC.
DELIVER_DATE	TIMESTAMP	Time when a message was sent from SMSC (null if message failed to deliver).
SERVICE_TYPE	TEXT	SMPP parameter (service_type) for ESME originated messages.
ESM_CLASS	INT	Indicates Message Mode (Messaging Mode==Datagram, Forward or Store and Forward mode) and Message Type (MessageType==some flags including UDH indicator).
PROTOCOL_ID	INT	Protocol Identifier SMPP parameter (TP-Protocol-Identifier files for GSM).
PRIORITY	INT	SMPP parameter (priority_flag).
REGISTERED_DELIVERY	INT	SMPP parameter (registered_delivery).
REPLACE	INT	SMPP parameter (replace_if_present_flag).
DATA_CODING	INT	data_coding scheme.
DEFAULT_MSG_ID	INT	SMPP parameter (sm_default_msg_id).
MESSAGE	BLOB	Message text in source style that has been received from EMSE or from MS.
OPTIONAL_PARAMETERS	TEXT	TLVs.

Column Name	Data Type	Description
SCHEDULE_DELIVERY_TIME	TIMESTAMP	SMPP parameter (schedule_delivery_time) - time when SMSC should start a delivery (may be null if immediate message delivery).
VALIDITY_PERIOD	TIMESTAMP	The validity period of this message. If ESME has not defined (or for MO messages) this field is filled by default SMSC settings.
IMSI	ASCII	From SRI response.
CORR_ID	ASCII	This field is used for keeping of correlationId value for home routing mode. This table will keep this value for logging purpose.
NNN_DIGITS	ASCII	NetworkNodeNumber = MSC that serves MS – from SRI response.
NNN_AN	INT	
NNN_NP	INT	
SM_STATUS	INT	Error Code value for the last attempt (0==no attempts yet). For more details on Error Codes please refer to Appendix A, Error Codes
SM_TYPE	INT	0-ESME terminated, 1-MT
DELIVERY_COUNT	INT	Delivery attempt count. (this will be==1 if a message was delivered in one go)

3.1.7. Reporting

As of now there is no reporting in &THIS.PLATOFORM;SMSC . However you can leverage any external third party tool to dig Cassandra tables as defined above and create reports. The next release of &THIS.PLATOFORM;SMSC will feature a reporting section and also the flexibility to search for a specific SMS based on various search criteria.

Chapter 4. Running

4.1. Running the Gateway

Procedure: Run Restcomm SMSC

1. Pre-requisite:
   - You must have Restcomm SMSC 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.
   - You must have Cassandra database running and configured as explained in the Installation Guide.
   - You must have the Cassandra Keyspace created as explained in the Installation Guide.
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 *restcomm-smscgateway-<version>/jboss-5.1.0.GA/bin* folder (on Unix or Windows). Note that this will start the server in the default profile. The "default" profile is a clean profile where you will have to start from scratch and configure the entire SS7 Stack and SMSC Gateway.
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:

```
2013-07-22 17:41:48,817 INFO
[org.mobicents.slee.container.management.ResourceManagement] (pool-27-thread-1)
Created Resource Adaptor Entity SmppServerRA for
ResourceAdaptorID[name=SMPPServerResourceAdaptor,vendor=org.mobicents,version=1.0]
Config Properties: []
2013-07-22 17:41:49,067 INFO
[org.mobicents.slee.container.management.ResourceManagement] (pool-27-thread-1)
Activated RA Entity SmppServerRA 2013-07-22 17:41:49,317 INFO
[org.mobicents.slee.container.management.ResourceManagement] (pool-27-thread-1)
Bound link between RA Entity SmppServerRA and Name SmppServerRA
2013-07-22 17:41:49,615 INFO
[javax.slee.RAEntityNotification[entity=SchedulerResourceAdaptor].SchedulerResource
Adaptor] (pool-27-thread-1) Verify configuration in RA Entity
SchedulerResourceAdaptor
2013-07-22 17:41:49,616 INFO
[org.mobicents.slee.container.management.ResourceManagement] (pool-27-thread-1)
Created Resource Adaptor Entity SchedulerResourceAdaptor for
ResourceAdaptorID[name=SchedulerResourceAdaptor,vendor=org.mobicents,version=1.0]
Config Properties: []
2013-07-22 17:41:49,866 INFO
[org.mobicents.slee.container.management.ResourceManagement] (pool-27-thread-1)
Activated RA Entity SchedulerResourceAdaptor
.
.
```

```
.
2013-07-22 17:41:51,122 INFO
[org.mobicents.slee.container.management.jmx.DeploymentMBeanImpl] (pool-27-thread-
1) Installing DeployableUnitID[url=file:/C:/JavaT/jboss/server/default/deploy/smsc-
services-du-6.1.2-RestComm-SNAPSHOT.jar/]
2013-07-22 17:41:52,307 INFO
[org.mobicents.slee.container.management.jmx.DeploymentMBeanImpl] (pool-27-thread-
1) Installed LibraryID[name=org.mobicents.smsc,vendor=smsc-library,version=1.0]
2013-07-22 17:41:52,349 INFO
[org.mobicents.slee.container.management.jmx.DeploymentMBeanImpl] (pool-27-thread-
1) Installed SbbID[name=AlertSbb,vendor=org.mobicents,version=1.0]
2013-07-22 17:41:52,408 INFO
[org.mobicents.slee.container.management.jmx.DeploymentMBeanImpl] (pool-27-thread-
1) Installed SbbID[name=RxSmppServerSbb,vendor=org.mobicents,version=1.0]
2013-07-22 17:41:52,618 INFO
[org.mobicents.slee.container.management.jmx.DeploymentMBeanImpl] (pool-27-thread-
1) Installed SbbID[name=SriSbb,vendor=org.mobicents,version=1.0]
.
.
.
2013-07-22 17:41:53,290 INFO
[org.mobicents.slee.container.management.jmx.DeploymentMBeanImpl] (pool-27-thread-
1) Installed DeployableUnitID[url=file:/C:/JavaT/jboss/server/default/deploy/smsc-
services-du-6.1.2-RestComm-SNAPSHOT.jar/]
2013-07-22 17:41:53,543 INFO
[org.mobicents.slee.container.management.ServiceManagementImpl] (pool-27-thread-1)
Activated ServiceID[name=MoService,vendor=org.mobicents,version=1.0]
.
.
.
 2013-07-22 17:41:54,974 INFO
[me.prettyprint.cassandra.connection.CassandraHostRetryService] (Thread-27) Downed
Host Retry service started with queue size -1 and retry delay 10s
2013-07-22 17:41:55,004 INFO [me.prettyprint.cassandra.service.JmxMonitor] (Thread-
27) Registering JMX
me.prettyprint.cassandra.service_RestCommSMSC:ServiceType=hector,MonitorType=hector
2013-07-22 17:41:55,014 INFO
[javax.slee.RAEntityNotification[entity=SchedulerResourceAdaptor].SchedulerResource
Adaptor] (Thread-27) Scheduler IS up, starting fetch tasks
2013-07-22 17:41:55,019 INFO [org.mobicents.smsc.smpp.SmscManagement] (main) SMSC
configuration file path C:\JavaT\jboss\server\default\data\SmscManagement_smsc.xml
2013-07-22 17:41:55,022 INFO [org.mobicents.smsc.smpp.SmppServerManagement] (main)
Loading SMPP Server Properties from
C:\JavaT\jboss\server\default\data\SmscManagement_smppserver.xml
2013-07-22 17:41:55,297 INFO [org.mobicents.smsc.smpp.SmppServerManagement] (main)
Starting SMPP server...
2013-07-22 17:41:55,322 INFO [com.cloudhopper.smpp.impl.DefaultSmppServer] (main)
SmscManagement started on SMPP port [2776]
2013-07-22 17:41:55,322 INFO [org.mobicents.smsc.smpp.SmppServerManagement] (main)
SMPP server started
2013-07-22 17:41:55,378 INFO [org.mobicents.smsc.smpp.SmppClientOpsThread] (Thread-
```

```
30) SmppClientOpsThread started.
2013-07-22 17:41:55,379 INFO [org.mobicents.smsc.smpp.SmscManagement] (main)
Started SmscManagement
2013-07-22 17:41:55,379 INFO
[javax.slee.RAEntityNotification[entity=SmppServerRA].SmppServerResourceAdaptor]
(main) Activated RA Entity SmppServerRA
2013-07-22 17:42:00,652 INFO [org.apache.coyote.http11.Http11Protocol] (main)
Starting Coyote HTTP/1.1 on http-127.0.0.1-8080
2013-07-22 17:42:00,674 INFO [org.apache.coyote.ajp.AjpProtocol] (main) Starting
Coyote AJP/1.3 on ajp-127.0.0.1-8009
2013-07-22 17:42:00,689 INFO [org.jboss.bootstrap.microcontainer.ServerImpl] (main)
JBoss (Microcontainer) [5.1.0.GA (build: SVNTag=JBoss_5_1_0_GA date=200905221634)]
Started in 1m:45s:372ms
```

4. If you are starting Restcomm SMSC for the first time, SS7 and SMSC working parameters (and possibly database access) are not configured. You need to use the Shell Client or the GUI to connect to the Gateway and configure the SS7 Stack and SMSC. Once configured, the state and configuration is persisted which stands server re-start. The next chapter will discuss in detail about configuration.

Procedure: Bind the SMSC Gateway to a specific IP

Using `run.sh` without any arguments binds the SMSC Gateway to localhost `127.0.0.1`. To bind the Gateway to a different IP, pass the IP address as value to the `-b` command line option. For example to bind the SMSC to `10.199.7.23` you will use the command from the *bin* folder as below:

```
[vsmsc bin]$ ./run.sh -b 10.199.7.23
```

Procedure: Stop the Gateway

1. To stop the Restcomm SMSC , 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 *restcomm-smscgateway-<version>/jboss-5.1.0.GA/bin* directory (on Unix or Windows).
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
```

4.2. Running the Gateway - Simulator Profile

The Restcomm SMSC 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, the smpp-simulator and some SIP phone. 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 SMSC Simulator profile is configured for use in Linux systems. For using it under Microsoft Windows you need to open the file *restcomm-smscgateway-<version>/jboss-5.1.0.GA/server/simulator/data/SCTPManagement_sctp.xml* and replace (in two places) `ipChannelType="0"` to `ipChannelType="1"` for using a TCP connection instead of SCTP (SCTP is not supported under Windows).

4.2.1. Configuring the Gateway in Simulator Profile

The SMSC Gateway in a Simulator Profile is configured as it would have been if configured with the following CLI commands:

```
sctp server create serv1 127.0.0.1 8012 sockettype SCTP
sctp server start serv1
sctp association create ass1 SERVER serv1 127.0.0.1 8011 sockettype SCTP

m3ua as create as1 IPSP mode SE ipspType server rc 101 traffic-mode loadsharing
network-appearance 102
m3ua asp create asp1 ass1
m3ua as add as1 asp1
m3ua asp start asp1
m3ua route add as1 1 2 3

sccp sap create 1 1 2 2
sccp dest create 1 1 1 1 0 255 255
sccp address create 1 82 1 8 0 1 4 000
sccp address create 2 82 2 8 0 1 4 000
sccp rule create 1 K 82 0 8 0 1 4 * solitary 1 origination-type localOriginated
sccp rule create 2 K 82 0 8 0 1 4 * solitary 2 origination-type remoteOriginated
sccp rsp create 1 1 0 0
sccp rss create 1 1 8 0

smsc set scgt 22220
smsc set scssn 8
smsc set hlrssn 8
smsc set mscssn 8
smsc set maxmapv 3

smpp esme create test test 127.0.0.1 -1 TRANSCEIVER SERVER password test esme-ton -1
esme-npi -1 esme-range 6666 source-range 6666 routing-range 6666 charging-enabled
false
smpp esme start test

smsc sip modify SIP cluster-name SIP host 127.0.0.1 port 5065 routing-ton 1 routing-
npi 1 routing-range 5555 counters-enabled false charging-enabled false
```

4.2.2. Running the jSS7 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.

Procedure: Running RestComm jSS7 Simulator

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

```
[vinu@vinu-neha ~]$ cd RestComm-smsc-<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` or in the case of Windows `./run.bat 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 SMSC Gateway (running in simulator profile). Press 'Run test' and again click on 'Start' in the next screen. The Simulator will connect to SMSC (via m3ua protocol).
5. The Low level part is configured to SCTP (not TCP) protocol and hence you can test the SMSC in a Linux environment. To test under Windows OS, you must change the SS7 simulator settings to TCP. The SS7 Simulator will play both HLR and MSC roles and respond to `SendRoutingInfo` and `ForwardSM` requests from SMSC gateways.
6. You can configure the `SMS_TEST_CLIENT` testing task to play with different modes including return error responses to SMSC and bulk mode without adding information for any event to an application form. You can also play with different data coding schemes and MAP protocol versions.

4.2.3. Running the SMPP Simulator

Procedure: Running SMPP Simulator

1. Change the working directory to the bin folder for the SMPP Simulator.

```
[vinu@vinu-neha ~]$ cd RestComm-smsc-<version>/tools/RestComm-smpp-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` or in the case of Windows `./run.bat`.

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

   This will launch the SMPP Simulator GUI Application.

4. The SMPP Simulator default settings fit to connect to the SMSC Gateway (running in simulator profile). You can in this case just press 'Run test' and then press on 'Start Session' to connect to the SMSC Gateway.
5. Using the button 'Configuring data for message submitting', you can configure different sending modes, data coding schemas, origination and destination addresses. See chapter SMPP Simulator Tool to understand more of SMPP simulator usage.
6. When the SMSC Gateway is running in a "simulator" profile it works with SMPP as ESME with the address "6666" (TON=1, NPI=1). So all messages with the destination address "6666" (TON=1, NPI=1) will be routed to SMPP Simulator. All other messages will be routed to SS7 Simulator (except messages for "5555" address that will be routed to SIP).

   With the SMPP Simulator you can also simulate a bulk message delivery to SMSC . This may be random bulk messages and bulk messages obtained from a pcap file.

4.2.4. Running the HLR Simulator

You can use the Command line HLR Simulator for load testing the SMSC . RestComm-hlr-simulator is pre-configured to integrate with SMSC run in simulator profile. The HLR Simulator receives the MAP SRI request from SMSC and returns response with random VLR and IMSI. SMSC will then forward `MT_Forward_SM` request to HLR. For every 7th `MT_Forward_SM` request, the HLR Simulator will return an `AbsentSubscriber` error.

For every 400 messages processed by the HLR Simulator, it shows the below message (time in milli seconds)

```
                Received 400 MAP Dialog requests in 1000
```

Procedure: Running HLR Simulator

1. Change the working directory to the bin folder for the HLR Simulator.

   ```
   [vinu@vinu-neha ~]$ cd RestComm-smsc-<version>/tools/RestComm-hlr-simulator/bin
   ```

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

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

3. Execute the *run.sh*

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

This will launch the HLR Simulator ready for processing SRI and MT_Forward_SM requests

4. The HLR Simulator is already pre-configured to connect to the SMSC Gateway (running in simulator profile).

Additional configuring of the HLR Simulator is possible only by manually updating the configuration files in the *RestComm-hlr-simulator/data* folder. For example, in order to run the HLR Simulator in Microsoft Windows you need to update the file *SCTPManagement_sctp.xml* and set the parameter `ipChannelType` to "1".

4.2.5. Running the SMPP Load tool

The smpp-load tool is a Command line simulator to generate SMPP load. You must have `ant` installed to be able to run this tool. The smpp-load tool can be started as a SMPP Server accepting in-coming connection (BIND) from Restcomm SMSC or can be started as a SMPP Client to send BIND to Restcomm SMSC . You can modify the configuration parameters in the *build.xml* to define how many SMPP connections should be initiated, what kind of load should be generated, etc.

Procedure: Running SMPP Load Tool

1. Change the working directory to the bin folder for the SMPP Load Tool.

```
[vinu@vinu-neha ~]$ cd RestComm-smsc-<version>/tools/RestComm-smpp-simulator/bin
```

2. Execute the `ant client` command to start the SMPP load tool as a client or execute the `ant server` command to start the SMPP load tool as a SMPP Server.

```
ant client
```

This will launch the SMPP load tool as a SMPP Client.

```
ant server
```

This will launch the SMPP load tool as a SMPP Server.

3. The SMPP load client is already pre-configured to connect to the Restcomm SMSC (running in simulator profile).

4.2.6. Running Jitsi or Linphone SIP phones

In order to test SMSC features for interconnection with SIP servers, you can use Jisti or Linphone SIP phones that can play the role of a SIP server.

Procedure: Configure Jitsi

1. Download Jisti from the website and run it.
2. Create a registrarless account (do not provide password, registrar and proxy address), enable PRESENSE (SIMPLE) at the forth tab for this account (Advanced). Identifier of this account must be "5555" because SMSC is configured to route all messages for subscriber "5555" to SIP.
3. In the menu, go to Tools → Options → Security → Chat, and enable all three options at the end.
4. Then go to Tools → Options → Advanced → SIP and define SIP port as "5065" (SMSC is configured for sending outgoing SIP messages to this port).
5. In the menu, go to File → Add contact and add a new contact with definition to which subscriber you will send messages to. SMSC is configured such that all messages for "6666" will be routed to ESME (SMPP simulator) and others (except "5555" that is for SIP) to GSM network (to SS7 Simulator). Therefore you can add two new contacts as below:
   - for SS7 Simulator: "sip:1111@127.0.0.1:5060"
   - for SMPP Simulator: "sip:6666@127.0.0.1:5060"

     The SMSC listens to 5060 port for incoming SIP messages. For addressing to/from the SS7 Simulator, the address "1111" will be used.

Procedure: Configure Linphone

1. Download Linphone from the website and run it.
2. Go to Parameters → Network settings → Network protocols and ports → SIP UDP port and set the port to 5066.
3. Add a contact with SIP address: "sip:6666@127.0.0.1:5060" for message exchange with SMPP simulator.
4. Add a contact with SIP address: "sip:1111@127.0.0.1:5060" for message exchange with SS7 simulator.

Now you can run the following tests:

1. For testing sending messages from a SIP phone you can use the feature "Send message" to a concrete contact, type a message text and send. After about 1 minute you will receive the message on the SS7 Simulator or the SIP Simulator.
2. For testing sending messages to the SIP phone you just need to send a message from the SS7 Simulator or SMPP Simulator to the address "5555". You must put the originating address as "1111" for SS7 Simulator and "6666" for SMPP Simulator.

4.3. 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-smscgateway-<version>/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=mobicents,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.

4.4. Running the Graphical User Interface

Open a Web Browser and navigate to http://IP:8080/mobicents-management/ (where IP is the IP Address to which the SMSC is bound to). This will launch the Restcomm GUI Management Console which is horizontally segregated into multiple tabs, one tab for each product in the Restcomm Suite. You will notice that only the tabs of products whose binaries are installed already will be shown enabled and active in the GUI. If you have successfully installed the Restcomm SMSC you will find the tabs for JAIN-SLEE, JMX, SS7 and SMSC GW active and enabled. For more details on using the GUI for SS7 or JAIN-SLEE please refer to their respective user guides. This document only provides instructions for using the GUI to configure the SMSC Gateway.

Switch to the SMSC GW tab and you will find that the window will look similar to the figure below. The GUI is divided into three sections:

- A left panel listing the management and monitoring units (Server Settings, SMPP Server, ESMEs, SIPs, MAP Version Cache, DB Routing Rules). You can click on any of these to select and navigate to the specific management unit.
- A main panel displaying the currently selected management unit. The main view is categorized into multiple tabs to manage different aspects of the selected layer.
- A bottom panel displaying the log data. You can clear the log anytime by clicking on the trash icon at the top right corner of this panel. You can also minimize or maximize this panel to suit your needs.

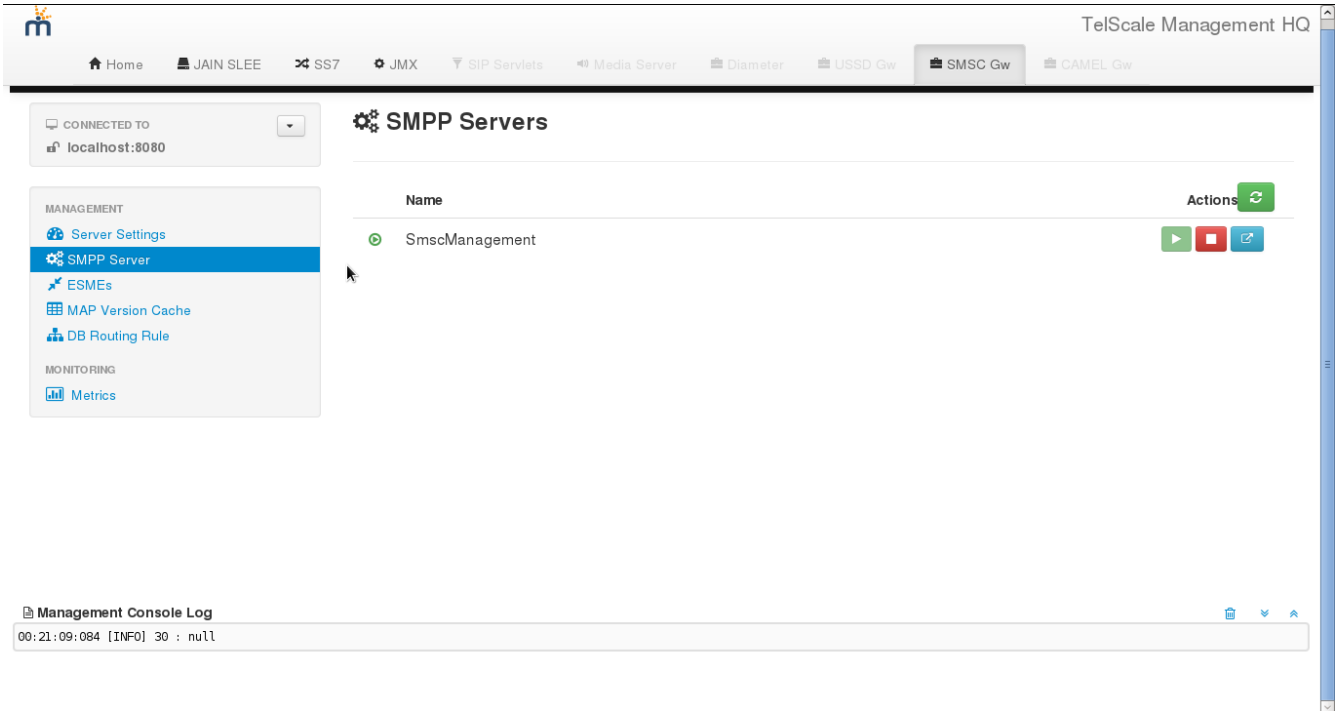


Figure 1. GUI - Restcomm SMSC

4.4.1. 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.

4.4.2. Authentication

Restcomm SMSC GUI Management Security is based on the JBoss Security Framework. To read more on JBoss Security Framework refer 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-smscgateway-<version>/jboss-5.1.0.GA/server/<profile>/conf/props/*

Chapter 5. Configuring

You must fine-tune Memory and Database settings for better performance before using Restcomm SMSC in production. Once you complete setting up the Gateway you must configure the SS7 Stack the SMSC Gateway following the instructions specified in this guide.

5.1. 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 *restcomm-smscgateway-version>/jboss-5.1.0.GA/bin/run.conf*.

-Xms3072m

Initial heap size, set in megabytes

-Xmx3072m

Maximum heap size, set in megabytes

5.2. Configuring log4j Logging Service

Restcomm SMSC uses `Apache log4j` for logging. If you are not familiar with the `log4j` package, you can read more about it at the Jakarta website.

Logging is controlled from a central configuration file located at *restcomm-smscgateway-<version>/jboss-5.1.0.GA/server/<profile>/conf/jboss-log4j.xml*, one for each JBoss AS configuration profile. This file defines a set of appenders specifying the log files, what categories of messages should go there, the message format and the level of filtering. For more details, please refer to Section 9.6.3, "Logging Service" in the JBoss AS Getting Started Guide available here.

You must make sure `log4j` is fine tuned for optimal performance in production. We recommend that you set logging threshold to `WARN` and let the CDR appender be `DEBUG`.

5.3. SMSC GW routing fundamentals

SMSC GW needs clear info to where it must route messages that have come to SMSC GW from SMPP / SS7 / SIP connectors or created inside SMSC GW receipts. Routing procedures gives the answer to this questions, that a user must configure before SMSC GW can properly work.

SMSC GW has two routing procedures:

- The procedure based on dividing of separated routing area into several areas ("subnetworks") and routing of messages between subnetworks.
- The procedure of routing messages to destinations inside subnetworks.

5.3.1. The procedure based on dividing of separated routing area into several areas ("subnetworks").

Each area has its unique digital "netwokId" value. Each SMPP connection (ESME), SS7 SCCP service access point (SAP), SIP connector and a database routing rule (if you use them instead of default ESME routing rules) belongs to one of network (to each of the a networkId value is assigned). Default netwokId value is "0". If a user does not specify a netwokId value when ESME, SAP, SIP or a database routing rule creation, networkId=0 will be assigned.

When a message has come to SMSC GW a networkId value from ESME, SAP or SIP is assigned for the message. The message can be routed only inside the area with the same networkId to what the message belongs. So this means that by default a message can not leave a "networkId" area via which it has come to SMSC GW.

If a user needs that a message will be delivered to another "networkId" area, he can configure "message processing" rules ("mproc" rules). There we can configure that a message from a specified networkId1 and which fits some other conditions (like destination address fit to some mask) must be moved to networkId2 (newNetworkId option of a mproc rule). And the message will be delivered via ESME, SAP or SIP that belongs to networkId2.

Multi-tenancy support is also based on this routing procedure. For each networkId we can configure a separate SS7 GlobalTitle and for an external peer each networkId can play a role as a separate independed SMSC.

NetworkId for a generated inside SMSC receipt is assigned by two algoritms:

- if the option "Delivery receipts will be routed to the origination networkId" is set to true - networkId from ESME/SIP/SAP via what an original message has left SMSC GW will be assigned
- if the option "Delivery receipts will be routed to the origination networkId" is set to false - networkId from ESME/SIP/SAP via what an original message has come to SMSC GW will be assigned

NetworkId are specified at different places:

- for JSS7 level SAPs and SCCP routing rules are configured for this. See "7.4.5. Create a new Service Access Point" and "7.4.17. Create a new SCCP Rule" chapters of "JSS7 stack User Guide".
- for ESMEs see the chapter External Short Messaging Entities (ESMEs).
- for SIP level see the chapter SIP Settings.
- for database routing rules see Database Routing Rules

For configuring of mproc rules - see chapters Message processing rules (mproc rules) and Message processing rules (mproc rules).

For configuring of option "Delivery receipts will be routed to the origination networkId" - see the chapter Routing of delivery receipts.

For configuring of SMSC Global Titles - see chapters SMSC Global Title.

5.3.2. The procedure of routing messages to destinations inside subnetworks.

Inside each subnetwork there may be several ESME, SAP or SIP configured. If we use default routing rules, then for routing SMSC GW makes following steps:

- it checks all ESMEs that belongs to the networkId. It checks if TON, NPI and AddressRange of an ESME Routing Address fit to a message destination TON, NPI and address digits the the message will be routed to this ESME. There is one exception of this rule. A message will not be routed in any case to the ESME from which it has come.
- if no ESME found for routing SMSC GW then checks SIP connector if it's settings (TON, NPI and routing AddressRange) fits to a message destination TON, NPI and address digits. If yes the message will be routed to SIP
- if SIP also does not fit, a message will be routed to SS7 network. Then routing rules at SCCP level will be taken into account for further routing.

For configuring of ESME and SIP routing parameters see External Short Messaging Entities (ESMEs) and SIP Settings.

Instead of default routing rules database routing rules can be used. This is a set of stored in the cassandra database rules that describes where SMSC GW will route a message (to some ESME / SIP) depending on a message destination address. Messages that fit to no rules will be routed to SS7 network. For configuring of database routing rules see the chapter Message processing rules (mproc rules)| |

5.3.3. Using of different routing procedures.

You can use only the procedure based on subnetworks (networkId) areas, only the procedure of routing messages inside subnetworks or both.

If you want to use only the procedure based on subnetworks (networkId) areas you need:

- configure SMSC GW so that only one SAP / ESME / SIP belongs to each networkId.
- for every ESME / SIP specify for "Routing Type of number (TON)"=-1, "Routing Number plan indicator (NPI)"=-1, "Routing Range"="^[0-9a-zA-Z]*" (that fits to any message destination address).
- configure a set of mproc rules that will manage of routing of messages from one subnetwork (networkId) to another.| |

If you want to use only the procedure based on subnetworks (networkId) areas you need:

- configured for all SAP / SCCP routing rules / ESME / SIP the same networkId (default networkId=0 is usually used)
- for every ESME / SIP specify proper values for "Routing Type of number (TON)", "Routing Number plan indicator (NPI)" and "Routing Range" or configure database routing rules
- you do not need to configure mproc rules for routing| |

If you want to use both, you need to configure both parts. Messages will be routed between subnetworks (networkIds) by mproc rules and inside a subnetwork (networkId) by configuring of ESME / SIP "Routing Type of number (TON)", "Routing Number plan indicator (NPI)" and "Routing Range" (or database routing rules).

5.3.4. Routing model examples.

If you have one ESME and JSS7 connector, and the only traffic is ESME → JSS7

- use networkId=0 for all
- configure "Routing Range" value of the ESME to a value that fits to no mobile subscriber (for example "0")

If you use the model "all JSS7 originated messages must be routed to ESME and all ESME originated messages must be routed to JSS7", you can:

- set networkId=0 for JSS7 and networkId=1 for ESME
- set ESME routing address so it accepts all messages
- set mproc rules:

  smsc mproc add mproc 1 networkIdMask=1 newNetworkId=0

  smsc mproc add mproc 2 networkIdMask=0 newNetworkId=1

If you have several ESMEs that send message to one SS7 connection and you need that delivery receipts come back to an originator ESME (and no more traffic), you can:

- put SS7 SAP / SCCP rules to networkId=0
- put each ESME to each own networkId (1,2,3)
- create a set of mproc rules that move ESME originated messages into SS7 network. CLI commabd can be like:

  smsc mproc add mproc 1 networkIdMask=1 newNetworkId=0

  smsc mproc add mproc 2 networkIdMask=2 newNetworkId=0

  smsc mproc add mproc 3 networkIdMask=3 newNetworkId=0

- set the option "Delivery receipts will be routed to the origination networkId" is set to true

5.4. Message processing rules (mproc rules)

Message processing rules (mproc rules) is a tool for processing messages and changing properties of message, for example source/destination TON, NPI, NetworkId etc. MProc rules are only applied if pre configured criterian match’s.

Following are the states at which mproc rules can be applied to messages:

1. `onPostArrival` : When a message arrives in SMSC GW and has been already processed (accepted) by Diameter server (if Diameter server is configured). `onPostArrival` following actions are possible:
    - message can be dropped (a success response will be returned to a message originator)
    - message can be rejected (a reject response will be returned to a message originator)
    - most of a message parameters can be updated (for example destination address, networkId (this is needed for routing such messages to another subnetwork area (networkId) (see chapter SMSC GW routing fundamentals)) or even a message content)
    - a new message(es) can be posted for delivering. To post new messages at this step no Diameter server request and no mproc rules will be applied

Only after actions applying messages will be stored into a database or/and delivered to a destination.

- `onPostImsiRequest` : When a successful SRI response has been received from HLR for an SS7 destination message. At this step a message can be dropped (to prevent further delivery). This can be very useful feature to prevent SMS being delivered to off-net or roaming subscribers.
- `onPostDelivery` : When a message delivery was succeeded or failed. At this step a new message(es) can be posted for delivering (it can be for example some delivery report). | |

SMSC comes with predifined set of mproc rules (default implementation) (see chapter Default message processing rules implementation). However users can make their own customized implementation of mproc rules by using java programming and implementing provided interfaces (see chapter Customized message processing rules). All mproc rules implementation has its Class Name. The Class Name of a default implementation is "mproc". This is the name by which users can create new mproc rule instances.

Users can create one or more mproc rules, modify, show and remove some of them by CLI or GUI interface. See details in chapter Message processing rules (mproc rules). Each mproc rule has it's unique serial id. Mproc rule are sorted by this id value. SMSC appliec mproc rules to a message in ascending order, that is mproc rule with the least id is applied before mproc rule with next id etc.

While checking if mproc rule conditions match to a message, updates (that were made after previous rules applying) are taken into account. For example if a message destination address has been changed by rule 1, then rule 2 will check if this updated destination address matches to rules 2 or not.

5.4.1. Default message processing rules implementation

SMSC GW contains a default implementation of mproc rules that cover some requirenments. Information how to manage rules can be found in chapter Message processing rules (mproc rules). This chapter covers a description of conditions and actions that are present in default mproc rules implementation.

Parameters for mproc rule are divided into two cathegories:

a) Conditions. If a message fits to all conditions then the rule will be applied to the message.

Table 7. The list of possible conditions

Parameter name	Value	Description	Default value
desttonmask	<destination type of number>	mproc rule will be applied only if message destination Type of Number is equal to this value "desttonmask"	"-1" : acts as wild card and hence messages with any TON will match.
destnpimask	<destination numbering plan indicator>	mproc rule will be applied only if message destination Numbering Plan Indicator is equal to this value "destnpimask".	"-1" : acts as wild card and hence messages with any NPI will match.
destdigmask	<java regular expression - destination number digits mask>	mproc rule will be applied only if message destination address digits match's with "destdigmask" java regular expression.	"-1" : acts as wild card and hence messages with any destination number will match.
originatingmask	<SS7_MO | SS7_HR | SMPP | SIP>	mproc rule will be applied only if message arrived in SMSC GW via a defined "originatingmask" connector. . SS7_MO: SMS Originated from Mobile - SS7 connection . SS7_HR: SMS Originated from Home Routing configuartion - SS7 connection . SMPP: SMS Originated from SMPP connector and SIP connector. . SIP: SMS Originated from SIP connector.	"-1" : acts as wild card and hence messages originated from any channel will match.
originatorsccpaddress-mask	<java regular expression - originator CallingPartyAddress digits mask>	mproc rule will be applied only if CallingPartyAddress digits match's with "originatorsccpaddress mask" java regular expression.	"-1" : acts as wild card and hence messages with any CallingPartyAddress digits or without it will match.

Parameter name	Value	Description	Default value
networkidmask	<networkId value>	mproc rule will be applied only if message has come to SMSC GW via a subnetwork with networkId which is equal to this value "networkidmask".	"-1" : acts as wild card and hence messages from any network will match.
origesmenamemask	<java regualar expression - origination ESME name mask>	mproc rule will be applied only if message has come to SMSC GW from SMPP connector from ESME with a name that fits "origesmenamemask".	"-1" : acts as wild card and hence any message will match (never mind if it came from SS7 or SIP).

b) Actions, which will be applied to messages.

Table 8. The list of possible actions

Parameter name	Value	Description	Default value
newnetworkid	<new networkId value>	networkId of the message will be changed to "newnetworkid" value. This means that the message will be delivered via connectors that belong to the new networId.	"-1". This means that networkId of the message will not be changed.
newdestton	<new destination type of number>	a message destination Type of Number will be changed to "newdestton" value.	"-1". This means that destination Type of Number of the message will not be changed.
newdestnpi	<new destination numbering plan indicator>	a message destination Numbering Plan Indicator will be changed to "newdestnpi" value.	"-1". This means that destination Numbering Plan Indicator of the message will not be changed.

Parameter name	Value	Description	Default value
adddestdigprefix	<prefix>	adddestdigprefix will be added into a begin of a message destination address digits. For example if adddestdigprefix is "22" and destination address digits are "3333333", then the new value of destination address digits will be "223333333".	"-1". This means that destination address digits of the message will not be changed.
makecopy	<false | true>	If the makecopy action is present then SMSC GW makes a copy of a message and then post the copy for further message processing in addition to the original message. All other actions in the rule will be applied only to the copy. For example user wants to make copies of messages and send them to another destinations (by sending of copies into another networkId), then for this you can create a rule and set for the rule "makecopy true" and "newnetworkid <new networkId value>" parameters. This makes a copy of a message and set for the copy a new networkId value.	false

Parameter name	Value	Description	Default value
dropaftersri	<false | true>	This is an action at the step when a successful SRI response has been received from HLR for an SS7 destination message. If a SRI success response has received then the message will be dropped without delivery attempt. A delivery response in this case will contain extra fields (IMSI and NetworkNodeNumber values).	false

5.4.2. Customized message processing rules

Default mproc rules allows to change the properties of a message in pre defined way, however if user wants to achieve more, SMSC allows users to implement their own custom logic. Below steps describes how to implement custom mproc rules.

Procedure: Steps for custom mproc rules implementing

1. User should implement the business logic as java code. There are couple of interfaces exposed by SMSC, `MProcRuleFactory` and `MProcRule` that must be implemented and add the custom business logic. User needs to cover two parts of rules usage - a rule configuring part and a rule applying part.

   Please pay attention that in your code you may not perform long delays in order not to dramatically decrease of SMSC GW productivity.

   Once custom rule is implemented, user will have to create jar file and deploy it into SMSC Gateway.

   - User needs to decide a rule class name that will be used to uniquely identify the custom rules. This can be any word without spaces. For default mproc rules implementation the rule class name is "mproc". For example consider rule class name as "testrule".
   - User needs to impelent two interfaces: `MProcRuleFactory` and `MProcRule`. For example consider `MProcRuleFactoryTestImpl` and `MProcRuleTestImpl`.
   - User needs to decide which actions will custom rules perform and for which messages. For example create a custom rule that will be applied to any message at `onPostArrival` state who's destination address digits are starting with a configurable parameter "par1". For all the messages which match's this rules condition, prefix "par2" (the configurable parameter) is to be applied to message destination address.

2. Creating custom classes:

```
package org.mobicents.smsc.mproc.testimpl;

import org.mobicents.smsc.mproc.MProcRuleFactory;

public class MProcRuleFactoryTestImpl implements MProcRuleFactory {
}
```

```
package org.mobicents.smsc.mproc.testimpl;

import org.mobicents.smsc.mproc.MProcRule;

public class MProcRuleTestImpl implements MProcRule {
}
```

3. Implementing `MProcRuleFactory` interface. The interface is:

```
package org.mobicents.smsc.mproc;

public interface MProcRuleFactory {
    String getRuleClassName();
    MProcRule createMProcRuleInstance();
}
```

Method `getRuleClassName()` must return the rule class name. Method `createMProcRuleInstance()` must return a custom implementation of `MProcRule` (in this example instance of `MProcRuleTestImpl` class). Here is an example of implementation:

```
package org.mobicents.smsc.mproc.testimpl;

import org.mobicents.smsc.mproc.MProcRule;
import org.mobicents.smsc.mproc.MProcRuleFactory;

public class MProcRuleFactoryTestImpl implements MProcRuleFactory {
    public static final String CLASS_NAME = "testrule";

    @Override
    public String getRuleClassName() {
        return CLASS_NAME;
    }

    @Override
    public MProcRule createMProcRuleInstance() {
        return new MProcRuleTestImpl();
    }

}
```

4. Next is actual business logic that should go in implement of `MProcRule` interface. Let's start with learning of the interface. The content of the interface is the following:

```
package org.mobicents.smsc.mproc;

public interface MProcRule extends MProcRuleMBean {

    void setId(int val);

    /**
     * @return true if the mproc rule fits to a message when a message has just
come to SMSC
     */
    boolean matchesPostArrival(MProcMessage message);

    /**
     * @return true if the mproc rule fits to a message when IMSI / NNN has been
received from HLR
     */
    boolean matchesPostImsiRequest(MProcMessage message);

    /**
     * @return true if the mproc rule fits to a message when a message has just
been delivered
     * (or delivery failure)
     */
    boolean matchesPostDelivery(MProcMessage message);

    /**
     * the event occurs when a message has just come to SMSC
     */
    void onPostArrival(PostArrivalProcessor factory, MProcMessage message) throws
Exception;

    /**
     * the event occurs when IMSI / NNN has been received from HLR
     */
    void onPostImsiRequest(PostImsiProcessor factory, MProcMessage message) throws
Exception;

    /**
     * the event occurs when a message has just been delivered (or delivery
failure)
     */
    void onPostDelivery(PostDeliveryProcessor factory, MProcMessage message) throws
Exception;

    /**
     * this method must implement setting of rule parameters as for provided CLI
string at
```

```
     * the step of rule creation
     */
    void setInitialRuleParameters(String parametersString) throws Exception;

    /**
     * this method must implement setting of rule parameters as for provided CLI
string at
     * the step of rules modifying
     */
    void updateRuleParameters(String parametersString) throws Exception;

}
```

and the parent interface content is:

```
package org.mobicents.smsc.mproc;

public interface MProcRuleMBean {

    /**
     * @return the id of the mproc rule
     */
    int getId();

    /**
     * @return Rule class of the mproc rule ("default" or other when a customer
implementation)
     */
    String getRuleClassName();

    /**
     * @return true if the mproc rule is used for the phase when a message has just
come to SMSC
     */
    boolean isForPostArrivalState();

    /**
     * @return true if the mproc rule is used for the phase when IMSI / NNN has
been received
     * from HLR
     */
    boolean isForPostImsiRequestState();

    /**
     * @return true if the mproc rule is used for the phase when a message has just
been
     * delivered (or delivery failure)
     */
    boolean isForPostDeliveryState();

    /**
     * @return rule parameters as CLI return string
     */
    String getRuleParameters();

}
```

Table 9. `MProcRule` and `MProcRuleMBean` interfaces methods description

Methods	Description
getId(), setId()	mproc rule id getter and setter. id value is unique for each mproc rule inside SMSC GW.
getRuleClassName()	Returns the rule class name ("testrule" in this example)

Methods	Description
getRuleParameters(), setInitialRuleParameters(), updateRuleParameters()	Getter and setters of configured parameters for a rule instance. Parameters are formed as a plain text string, that can be provided by CLI interface. In this example we need to configure two parameters par1 and par2. Let's specify the parameters strinf as "par1" and "par2" (two values after a space, for example "11 34").
isForPostArrivalState(), isForPostImsiRequestState(), isForPostDeliveryState()	Returns true if this rule affects to a message processing step (for onPostArrival, onPostDelivery and onPostImsi requests) and false if not. You have to check configured parameters of a rule and return a proper value
matchesPostArrival(MProcMessage message), matchesPostImsiRequest(MProcMessage message), matchesPostDelivery(MProcMessage message)	These methods must return true if rule should be applied false if not (for onPostArrival, onPostImsi and onPostDelivery requests).
onPostArrival(), onPostImsiRequest(), onPostDelivery()	These methods are needed for implementing of rule applying actions.

Id parameter and custom rule parameters ("par1" and "par2" in this example) must be stored into xml config file (it is located in *restcomm-smscgateway-<version>/jboss-5.1.0.GA/server/<profile>/data/SmscManagement_mproc.xml*). For this user needs to implement several extra methods that will be described later.

5. Reusing of base `MProcRuleBaseImpl` class. It is recommended for your custom `MProcRuleTestImpl` to extend `MProcRuleBaseImpl` provided by SMSC. `MProcRuleBaseImpl` class contains some useful methods that cover getter and setter and code for persisting of this parameter into xml config file (read() and write() methods), default stubs for `matches()` and `onPost*()` methods and a service method for splitting of a parameters plain text string into subparameters (splitParametersString()).

   Here is the new template of `MProcRuleTestImpl` class implementation that reuses the base class `MProcRuleBaseImpl`:

   ```
   package org.mobicents.smsc.mproc.testimpl;

   import org.mobicents.smsc.mproc.MProcRuleBaseImpl;

   public class MProcRuleTestImpl extends MProcRuleBaseImpl {
   }
   ```

6. Implementing of `getRuleClassName()` method of `MProcRule`. Below is an example:

```
    @Override
    public String getRuleClassName() {
        return MProcRuleFactoryTestImpl.CLASS_NAME;
    }
```

7. Implementing of methods that cover getter / setters for message custom parameters and storing them into xml config file.

   In the example setInitialRuleParameters() and updateRuleParameters() methods are implemented in the same way. You can implement them in different ways so that updateRuleParameters() method can accept not the whole set of parameters but only that ones that a user wants to change.

   M_PROC_RULE_TEST_XML is responsible for XML serializing / deserializing. See more info for them in javolution library specification.

   Here is an example (only the part that is described in this step):

```
package org.mobicents.smsc.mproc.testimpl;

import javolution.xml.XMLFormat;
import javolution.xml.stream.XMLStreamException;
import org.mobicents.smsc.mproc.MProcRuleBaseImpl;

public class MProcRuleTestImpl extends MProcRuleBaseImpl {

    private static final String PAR1 = "par1";
    private static final String PAR2 = "par2";
    private String par1, par2;

    @Override
    public void setInitialRuleParameters(String parametersString) throws Exception
{
        String[] args = splitParametersString(parametersString);
        if (args.length != 2) {
            throw new Exception("parametersString must contains 2 parameters");
        }
        par1 = args[0];
        par2 = args[1];
    }

    @Override
    public void updateRuleParameters(String parametersString) throws Exception {
        String[] args = splitParametersString(parametersString);
        if (args.length != 2) {
            throw new Exception("parametersString must contains 2 parameters");
        }
        par1 = args[0];
        par2 = args[1];
```

```
    }

    @Override
    public String getRuleParameters() {
        return par1 + " " + par2;
    }

    /**
     * XML Serialization/Deserialization
     */
    protected static final XMLFormat<MProcRuleTestImpl> M_PROC_RULE_TEST_XML = new
            XMLFormat<MProcRuleTestImpl>(MProcRuleTestImpl.class) {

        @Override
        public void read(javolution.xml.XMLFormat.InputElement xml,
MProcRuleTestImpl
            mProcRule) throws XMLStreamException {
            M_PROC_RULE_BASE_XML.read(xml, mProcRule);

            mProcRule.par1 = xml.getAttribute(PAR1, "");
            mProcRule.par2 = xml.getAttribute(PAR2, "");
        }

        @Override
        public void write(MProcRuleTestImpl mProcRule, javolution.xml.XMLFormat
.OutputElement
            xml) throws XMLStreamException {
            M_PROC_RULE_BASE_XML.write(mProcRule, xml);

            xml.setAttribute(PAR1, mProcRule.par1);
            xml.setAttribute(PAR2, mProcRule.par2);
        }
    };
}
```

8. Overiding one of `isForPost*()` methods (for allowing of processing of a needed message processing step). In our example it is `isForPostArrivalState()`:

```
    @Override
    public boolean isForPostArrivalState() {
        return true;
    }
```

9. Implementing of a needed `matches***()` methods. In our example we process messages which destination address digits are started with a configurable parameter "par1".

```
    @Override
    public boolean matchesPostArrival(MProcMessage message) {
        if (message.getDestAddr().startsWith(par1))
            return true;
        else
            return false;
    }
```

For `matches***()` methods we will use interface `MProcMessage` interface which provides info for processing message fields. Here is a code of these interfaces.

```
package org.mobicents.smsc.mproc;

import java.util.Date;

public interface MProcMessage {

    // source address part
    int getSourceAddrTon();

    int getSourceAddrNpi();

    String getSourceAddr();

    // dest address part
    int getDestAddrTon();

    int getDestAddrNpi();

    String getDestAddr();

    // message content part
    String getShortMessageText();

    byte[] getShortMessageBin();

    // other options
    int getNetworkId();

    int getOrigNetworkId();

    String getOrigEsmeName();

    OrigType getOriginationType();

    Date getScheduleDeliveryTime();

    Date getValidityPeriod();
```

```
    int getDataCoding();

    int getNationalLanguageSingleShift();

    int getNationalLanguageLockingShift();

    int getEsmClass();

    int getPriority();

    int getRegisteredDelivery();

    String getOriginatorSccpAddress();

}
```

10. Implementing of methods that make some actions: In example above this is `onPostArrival()` and the action is adding "par2" prefix into destination address digits. These methods will be invoked for all messages that match rule's conditions.

```
    @Override
    public void onPostArrival(PostArrivalProcessor factory, MProcMessage message)
throws Exception {
        String destAddr = this.par2 + message.getDestAddr();
        factory.updateMessageDestAddr(message, destAddr);
    }
```

In `onPostArrival()`, `onPostImsiRequest()` and `onPostDelivery()` methods user can use provided interfaces `PostArrivalProcessor`, `PostImsiProcessor` and `PostDeliveryProcessor` that provide methods for message processing / adding / dropping / rejecting.

- `PostArrivalProcessor` interface content:

```
package org.mobicents.smsc.mproc;

import java.util.Date;

import org.apache.log4j.Logger;

public interface PostArrivalProcessor {

    // access to environmental parameters
    /**
     * @return the logger that an application can use for logging info into
server.log
     */
    Logger getLogger();

    // actions
```

```
    /**
     * Drop the message. Success response (that a message is accepted) will be
return to
     * a message originator.
     */
    void dropMessage();

    /**
     * Drop the message. A reject will be sent to a message originator.
     */
    void rejectMessage();

    // updating of a message section
    void updateMessageNetworkId(MProcMessage message, int newNetworkId);

    /**
     * Updating of destination address message TON. In case of bad value (<0 or
>6)
     * MProcRuleException will be thrown
     *
     * @param message
     * @param newDestTon
     * @throws MProcRuleException
     */
    void updateMessageDestAddrTon(MProcMessage message, int newDestTon) throws
MProcRuleException;


    /**
     * Updating of destination address message NPI. In case of bad value (<0 or
>6)
     * MProcRuleException will be thrown
     *
     * @param message
     * @param newDestNpi
     * @throws MProcRuleException
     */
    void updateMessageDestAddrNpi(MProcMessage message, int newDestNpi) throws
MProcRuleException;

    /**
     * Updating of destination address message digits. Value can not be null and
must have length
     * 1-21 characters. In case of bad value MProcRuleException will be thrown
     *
     * @param message
     * @param newDigits
     * @throws MProcRuleException
     */
    void updateMessageDestAddr(MProcMessage message, String newDigits) throws
MProcRuleException;
```

```
    /**
     * Updating of source address message TON. In case of bad value (<0 or >6)
     * MProcRuleException will be thrown
     *
     * @param message
     * @param newDestTon
     * @throws MProcRuleException
     */
    void updateMessageSourceAddrTon(MProcMessage message, int newDestTon) throws
MProcRuleException;

    /**
     * Updating of source address message NPI. In case of bad value (<0 or >6)
     * MProcRuleException will be thrown
     *
     * @param message
     * @param newDestNpi
     * @throws MProcRuleException
     */
    void updateMessageSourceAddrNpi(MProcMessage message, int newDestNpi) throws
MProcRuleException;

    /**
     * Updating of source address message digits. Value can not be null and must
have length
     * 1-21 characters. In case of bad value MProcRuleException will be thrown
     *
     * @param message
     * @param newDigits
     * @throws MProcRuleException
     */
    void updateMessageSourceAddr(MProcMessage message, String newDigits) throws
MProcRuleException;

    /**
     * Updating of message text. Value must not be null and must have length 0-
4300. In case
     * of bad value MProcRuleException will be thrown
     *
     * @param message
     * @param newShortMessageText
     * @throws MProcRuleException
     */
    void updateShortMessageText(MProcMessage message, String
newShortMessageText) throws MProcRuleException;

    /**
     * Updating of UDH binary content. Value can be null or must have length >
0. In case
     * of bad value MProcRuleException will be thrown
```

```
     *
     * @param message
     * @param newShortMessageText
     * @throws MProcRuleException
     */
    void updateShortMessageBin(MProcMessage message, byte[] newShortMessageBin)
throws MProcRuleException;

    /**
     * Updating of ScheduleDeliveryTime - the time before which a message will
not be
     * delivered. This value can be null, this means that the message will be
tried to
     * delivery immediately. This value must be at least 3 hours before a
delivery
     * period end. If you pass the value that is later then 3 hours before a
delivery
     * period end, then 3 hours before a delivery period end will be set. If you
     * change both ValidityPeriod and ScheduleDeliveryTime values, then you have
to
     * setup ValidityPeriod value firstly.
     *
     * @param message
     * @param newScheduleDeliveryTime
     */
    void updateScheduleDeliveryTime(MProcMessage message, Date
newScheduleDeliveryTime);

    /**
     * Updating delivery period end time. This value can be null, this means
that
     * delivery period will be set to a default delivery period value of SMSC
GW.
     * If the value is more than max validity period that is configured for SMSC
GW,
     * then max validity period will be used instead of a provided value. If you
change
     * both ValidityPeriod and ScheduleDeliveryTime values, then you have to
setup
     * ValidityPeriod value firstly.
     *
     * @param message
     * @param newValidityPeriod
     */
    void updateValidityPeriod(MProcMessage message, Date newValidityPeriod);

    void updateDataCoding(MProcMessage message, int newDataCoding);

    void updateDataCodingGsm7(MProcMessage message);

    void updateDataCodingGsm8(MProcMessage message);
```

```
    void updateDataCodingUcs2(MProcMessage message);

    void updateNationalLanguageSingleShift(MProcMessage message,
        int newNationalLanguageSingleShift);

    void updateNationalLanguageLockingShift(MProcMessage message,
        int newNationalLanguageLockingShift);

    void updateEsmClass(MProcMessage message, int newEsmClass);

    void updateEsmClass_ModeDatagram(MProcMessage message);

    void updateEsmClass_ModeTransaction(MProcMessage message);

    void updateEsmClass_ModeStoreAndForward(MProcMessage message);

    void updateEsmClass_TypeNormalMessage(MProcMessage message);

    void updateEsmClass_TypeDeliveryReceipt(MProcMessage message);

    void updateEsmClass_UDHIndicatorPresent(MProcMessage message);

    void updateEsmClass_UDHIndicatorAbsent(MProcMessage message);

    void updatePriority(MProcMessage message, int newPriority);

    void updateRegisteredDelivery(MProcMessage message, int
newRegisteredDelivery);

    void updateRegisteredDelivery_DeliveryReceiptNo(MProcMessage message);

    void updateRegisteredDelivery_DeliveryReceiptOnSuccessOrFailure(MProcMessage
message);

    void updateRegisteredDelivery_DeliveryReceiptOnFailure(MProcMessage
message);

    void updateRegisteredDelivery_DeliveryReceiptOnSuccess(MProcMessage
message);

    // new message posting section
    /**
     * Creating a new message template for filling and sending by
postNewMessage() method
     */
    MProcNewMessage createNewEmptyMessage(OrigType originationType);

    MProcNewMessage createNewCopyMessage(MProcMessage message);

    MProcNewMessage createNewResponseMessage(MProcMessage message);
```

```
    /**
     * Posting a new message. To post a new message you need: create a message
template
     * by invoking of createNewMessage(), fill it and post it be invoking of
     * postNewMessage(). For this new message no mproc rule and diameter request
will
     * be applied.
     */
    void postNewMessage(MProcNewMessage message);
}
```

- `PostImsiProcessor` interface content:

```
package org.mobicents.smsc.mproc;

import org.apache.log4j.Logger;

public interface PostImsiProcessor {

    // access to environmental parameters
    /**
     * @return the logger that an application can use for logging info into
server.log
     */
    Logger getLogger();

    // actions
    void dropMessages();
}
```

- `PostDeliveryProcessor` interface content:

```
package org.mobicents.smsc.mproc;

import org.apache.log4j.Logger;

public interface PostDeliveryProcessor {

    // access to environmental parameters
    /**
     * @return the logger that an application can use for logging info into
server.log
     */
    Logger getLogger();

    boolean isDeliveryFailure();

    // actions
    /**
     * Creating a new message template for filling and sending by
postNewMessage() method
     */
    MProcNewMessage createNewEmptyMessage(OrigType originationType);

    MProcNewMessage createNewCopyMessage(MProcMessage message);

    MProcNewMessage createNewResponseMessage(MProcMessage message);

    /**
     * Posting a new message. To post a new message you need: create a message
template
     * by invoking of createNewMessage(), fill it and post it be invoking of
     * postNewMessage(). For this new message no mproc rule and diameter request
will
     * be applied.
     */
    void postNewMessage(MProcNewMessage message);

}
```

11. Once all the source code is fully implemented, user should deploy it.
    - User must compile java code and create a jar library. Copy the created jar into the folder *restcomm-smscgateway-<version>/jboss-5.1.0.GA/server/<profile>/deploy/restcomm-smsc-server/lib*.
    - You need to update *restcomm-smscgateway-<version>/jboss-5.1.0.GA/server/<profile>/deploy/restcomm-smsc-server/META-INF/jboss-beans.xml* config file.

      Define factory implementing `MProcRuleFactory`in above example it will be `MProcRuleFactoryTestImpl`:

```
    <bean name="MProcRuleFactoryTest"
class="org.mobicents.smsc.mproc.testimpl.MProcRuleFactoryTestImpl">
    </bean>
```

bean name can be some unique name.

In the "bean name="SmscManagement"" section - property <property name="MProcRuleFactories"> - after the line <inject bean="MProcRuleFactoryDefault"/> (that means default mproc rules implementing) - add the line that describes your new created bean:

```
        <property name="MProcRuleFactories">
            <list elementClass="org.mobicents.smsc.mproc.MProcRuleFactory">
                <inject bean="MProcRuleFactoryDefault"/>
                <inject bean="MProcRuleFactoryTest"/>
            </list>
        </property>
```

12. Start SMSC GW
13. Create an mproc rule by using CLI interface. Let's create a rule with id=1 that will be applied for messages which destination adresses are started with "22" and this rule will add prefix "33". To achive it run CLI console and run the following command

    smsc mproc add testrule 1 22 33

    where "testrule" is an implemented class name, "22 33" is aparameter string.

14. Send a message to the destination address that starts with "22" (for example "221111") and find that the message will be delivered to the address with prefix "33" ("33221111" in our example).| |

5.5. SMS Home Routing

According to traditional GSM specifications, all outbound messages pass through an SMSC in the sending network. This allows the SMSC to convert the Mobile Originated (MO) messages into Mobile Terminated (MT) messages and deliver them directly to the destination devices, regardless of what network they are on. As a result, inbound messages generated on other networks will be sent directly to the destination devices under the control of the SMSC in the sending network, not the home network. In this traditional setup, operators can add value to the MO messages but not to the MT messages that the customer may receive from other networks, since they do not pass through an SMSC in the home network.

SMS Home Routing is a modification to the original GSM specifications, adopted by the 3GPP in 2007, that changed the way inbound messages are treated by the mobile networks. Home Routing uses the recipient network's Home Location Register (HLR) to change the flow of inbound messages, directing them to an MT Services Platform, rather than straight to the destination

devices. The MT Services platform can add value to the MT messages and apply advanced services such as anti-spam, protection, divert, archiving, etc. to the messages prior to delivery.

Home Routing enables operators to offer a full range of value-added services to both inbound and outbound SMS thereby enhancing customer experience while generating additional revenue for the operators.

Restcomm SMSC supports SMS Home Routing that enables you to handle Mobile Terminated messages in the network that owns the customer so you can offer a full range of advanced and value added services on both inbound and outbound SMS. You need to configure SMSC GW before you can use SMS Home Routing. For more details refer to Home Routing Settings.

5.6. Interworking with Diameter OCS server

5.6.1. General information

SMSC GW can interoperate with an OCS server via Diameter protocol connection.

For this case SMSC GW can be configured so any incoming SMPP, MO SS7, Home Routing SS7 and SIP originated messages before next processing will be sent to OCS server. Then OCS server can check if SMSC GW must accept the message or not. If yes, SMSC GW will process the message, if not, the message will be dropped. OCS server can also make any charging operations as needed.

For SMPP and MO SS7 originated messages SMSC GW will return error (reject) in the response back to the orgiginated message submit message. So MO and SMPP part will be informed if the message is rejected. (This functionality is not implemented now for Home Routing SS7 and SIP originated messages).

For interconnecting with OCS Server SMSC GW uses diameter CCR (Credit-Control Request) with following filled AVPs (all of them are located in AVP Service-Information (873)):

Table 10. The AVP list

AVP name	AVP code	AVP Type	Parent AVP	Description
SMS-Information	2000	Grouped	Service-Information	SMS service specific information elements
Originator-SCCP-Address	2008	Address	SMS-Information	The CallingParty SCCP Address of MO message

AVP name	AVP code	AVP Type	Parent AVP	Description
SMSC-Address	2017	Address	SMS-Information	SMSC address as it is present in SM_RP_DA field in MO request. For not MO messages the value will be taken as a configured SMSC GT for networkId area
Data-Coding-Scheme	2001	Integer 32	SMS-Information	Data Coding Scheme of a message (for example 0 for GSM7 encoding or 8 for UCS2 encoding)
SM-Message-Type	2007			Message type, usually SUBMISSION (value = 0)
Originator-Interface	2009	Grouped	SMS-Information	Information related to the Interface on which the message originated.
Interface-Id	2003	UTF8-String	Originator-Interface	NetworkID value of the Interface on which the message originated.
Interface-Text	2005	UTF8-String	Originator-Interface	Name of ESME / SIP connector on which the message originated.
Recipient-Info	2026	Grouped	SMS-Information	Information associated with a recipient.
Recipient-Address	1201	Grouped	Recipient-Info	The address of message receiver (destination address)

AVP name	AVP code	AVP Type	Parent AVP	Description
Address-Type	899	Enum	Recipient-Address	Type of Recipient-Address: 1-MSISDN (for TON=1 - international) or 6-Other (for TON!=1)
Address-Data	897	UTF8-String	Recipient-Address	Digits of Recipient-Address
Originator-Received-Address	2027	Grouped	SMS-Information	The address of message sender (source address)
Address-Type	899	Enum	Originator-Received-Address	Type of Originator-Received-Address: 1-MSISDN (for TON=1 - international) or 6-Other (for TON!=1)
Address-Data	897	UTF8-String	Originator-Received-Address	Digits of Originator-Received-Address

When OCS server responds with 2001 code (access granted) SMSC GW will accept the message, all other respond codes will reject the message.

5.6.2. Configuring

You can use Telestax OCS server or any third party OCS server as you wish. SMSC GW party configuring is the same for both. Here there is a describtion for the case of Telestax OCS server and as an example - the simpliest case when we have OCS at the same host as SMSC GW.

Procedure: Steps for configuring of SMSC GW and OCS server

1. SMSC GW must be already configured for accepting and sending messages (general, SS7, SMPP, etc parts)
2. You can download for testing the Telestax OCS server from https://telestax.zendesk.com/forums/22947518-Product-Downloads. Download and unpack it to a separate folder.
3. SMSC GW uses the updated AVP dictionary library. Before using of OCS server copy from SMSC GW the file <smsc gw root folder>/resource/ocs/dictionary.xml to Telestax OCS server to the folder <jboss>/server/default/deploy/RestComm-diameter-mux-6.2.0.GA.sar/config.
4. for simple test configuring of SMSC GW and OCS server you can copy example config files jdiameter-config.xml

- from <smsc gw root folder>/resource/ocs/smsc-part to SMSC GW <jboss>/server/default/deploy/RestComm-diameter-mux-6.2.0.GA.sar/config
- from <smsc gw root folder>/resource/ocs/ocs-part to OCS server <jboss>/server/default/deploy/RestComm-diameter-mux-6.2.0.GA.sar/config

5. Those configs suppose that you use Telestax OCS server and it is on the same host. If not you need to make extra configuring. In the testing config files SMSC GW plays a role of a diameter client and OCS server - a diameter server. In this case for OCS server part we need to update jdiameter-config.xml file:
   - <LocalPeer> section with URI, IPAddresses and Realm.
   - In <Network> we need to specify <Peer> (SMSC GW diameter IP) and <Realm> (name and the same Realm that is configured in SMSC GW part)

     For SMSC GW part we need to update:

   - <LocalPeer> section with URI, IPAddresses and Realm.
   - In <Network> we need to specify <Peer> (OCS server diameter IP) and <Realm> (name and the same Realm that is configured in OCS server part) You can read more for configuring in the manuals for Telestax Diameter protocol and Telestax OCS server.
6. Start OCS server. If it is located in the same host as SMSC GW you need to run it "-b 127.0.0.2" parameter so it uses 127.0.0.2 IP address (127.0.0.1 is used by SMSC GW) Use fo starting the command like:

   ```
   run -b 127.0.0.2
   ```

7. For accessing OCS managing console (after OCS server is started) use a browser with URL: http://127.0.0.2:8080/charging-server-rest-management/#/users

   Avoid of using of Microsoft Internet Explorer, OCS server does not work correctly with it.

   Add one or more test subscribers by the console. You need to specify an address of message sender (source address). OCS Server idendifies a subscriber by its sender address.

8. Start SMSC GW
9. You need to configure Destination Realm, Destination Host, Destination Port and User Name (of OCS server) - see the chapter "6.1.5. Diameter Settings" of SMSC GW Admin guide. You can take values from jdiameter-config.xml. You can assign for User Name any value (for example "telestax").
10. You need to specify which messages will be sent to OCS server (see chapter Diameter Settings). For example if you want to charge mobile originated messages - specify "Mobile Originated SMS Charged" to value "diameter".
11. SMSC GW is configured

5.7. CDR Logging Settings

Restcomm SMSC is configured to generate CDR in a plain text file located at *restcomm-smscgateway-<version>/jboss-5.1.0.GA/server/<profile>/log/cdr.log* and also detailed CDR in Cassandra table `MESSAGES_yyyy_mm_dd`.

Restcomm SMSC can be configured to generate CDR for different message processing modes (`datagramm, transactional, storeAndForward`) and also for both receipt and regular messages or generate CDR only for regular messages. This is configurable by setting `generateCdr`, `generateArchiveTable` and `generateReceiptCdr` using the CLI or the GUI. For more details refer to Sections CDR generation, Archive table generation and Generate CDR for Receipt Messages.

The CDR generated in a text file is of a specific format. The details of the format and the possible values for the fields recorded in the CDR log file are explained in CDR Log.

Chapter 6. Managing

Now that you have set up the Gateway you must configure the SS7 Stack and the Gateway. &THIS.PLATFORM;&THIS.APPLICATION;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 SMSC Gateway dynamically. This chapter will explain how to manage the Gateway effectively using both the GUI and the CLI.

Configuring the SS7 Stack

You must configure the SS7 stack prior to configuring SMSC. 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.

Configuring the Gateway

Please find below relevant information on configuring the Gateway

6.1. SMSC Gateway Server Settings

6.1.1. SMSC Server Properties

Default Validity Period (in hours)

Using CLI

You can set the 'Default Validity Period (in hours)' by issuing the command `smsc set defaultvalidityperiodhours` with appropriate parameters as described below. You can verify this by issuing the command `smsc get defaultvalidityperiodhours` which will display the value set for this property.

```
Name
    smsc set defaultvalidityperiodhours

SYNOPSIS
    smsc set defaultvalidityperiodhours <default-validity-period-hours>

DESCRIPTION
    This command is used to set a value for default validity period (in hours) for
    incoming SMSC messages that do not have their own validity period value. Validity
    period is the time duration for which the SMSC Gateway will attempt to
    send the SMS. If the time period expires before the message can be delivered,
    the SMSC Gateway will drop it.

EXAMPLES
    smsc set defaultvalidityperiodhours 3
```

Using GUI

Procedure: Set Default Validity Period using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Properties' tab in the GUI.
3. You can specify the default validity period (in hours) by entering the value in the text field 'Default validity period hours (in hours)'. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

Maximum Validity Period (in hours)

Using CLI

You can set the 'Max Validity Period (in hours)' by issuing the command `smsc set maxvalidityperiodhours` with appropriate parameters as described below. You can verify this by issuing the command `smsc get maxvalidityperiodhours` which will display the value set for this property.

```
Name
    smsc set maxvalidityperiodhours

SYNOPSIS
    smsc set maxvalidityperiodhours <max-validity-period-hours>

DESCRIPTION
    This command is used to set a value for the maximum validity period (in hours).
    All incoming messages with a validity period set greater than the value specified
    by this parameter will either be rejected (if they are ESME originated messages)
    or the value of their validity period will be set to this value (for MO originated
    messages).

EXAMPLES
    smsc set maxvalidityperiodhours 6
```

Using GUI

Procedure: Set Maximum Validity Period using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Properties' tab in the GUI.
3. You can specify the maximum validity period (in hours) by entering the value in the text field

'Maximum validity period (in hours)'. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.

4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

Default Type of Number (TON)

Using CLI

You can set the 'Default type of number (TON)' by issuing the command `smsc set defaultton` with appropriate parameters as described below. You can verify this by issuing the command `smsc get defaultton` which will display the value set for this property.

```
Name
    smsc set defaultton

SYNOPSIS
    smsc set defaultton <default-ton>

DESCRIPTION
    This command is used to set a value for default TON (Type Of Number). If an
    incoming message does not have a value defined for TON, i.e. if TON is set as
    'unknown', then the value specified for defaultton will be used as TON for that
    message.

EXAMPLES
    smsc set defaultton 1
```

Using GUI

Procedure: Set Default type of number (TON) using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Properties' tab in the GUI.
3. You can specify the Default type of number (TON) by entering the value in the text field 'Default type of number (TON)'. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

Default numbering plan indicator (NPI)

Using CLI

You can set the 'Default numbering plan indicator (NPI)' value by issuing the command `smsc set defaultnpi` with appropriate parameters as described below. You can verify this by issuing the command `smsc get defaultnpi` which will display the value set for this property.

```
Name
    smsc set defaultnpi

SYNOPSIS
    smsc set defaultnpi <default-npi>

DESCRIPTION
    This command is used to set a value for default NPI (Number Plan Indicator). If an
    incoming message does not have a value defined for NPI, i.e. if NPI is set as
    'unknown', then the value specified for defaultnpi will be used as NPI for that
    message.

EXAMPLES
    smsc set defaultnpi 1
```

Using GUI

Procedure: Set Default numbering plan indicator (NPI) using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Properties' tab in the GUI.
3. You can specify the Default numbering plan indicator (NPI) by entering the value in the text field 'Default numbering plan indicator (NPI)'. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

Cluster Name

Using CLI

You can set the 'Cluster Name' value by issuing the command `smsc set esmedefaultcluster` with appropriate parameters as described below. You can verify this by issuing the command `smsc get esmedefaultcluster` which will display the value set for this property.

```
Name
    smsc set esmedefaultcluster

SYNOPSIS
    smsc set esmedefaultcluster <esme-default-cluster>

DESCRIPTION
    This command is used to set a value for ESME default cluster. If the
    destination-address does not match to any ESME (any Cluster Name) the message will
    be routed to the cluster with the name specified here for esme-default-cluster.
```

You can remove an ESME default cluster by issuing a command in the below format:

```
Name
    smsc remove esmedefaultcluster

SYNOPSIS
    smsc remove esmedefaultcluster <esme-default-cluster>

DESCRIPTION
    This command is used to remove the value configured for ESME default cluster. If
    this value is removed, all unrouted messages will be routed into the GSM network.
```

Using GUI

Procedure: Set Cluster Name using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Properties' tab in the GUI.
3. You can specify the Cluster Name by entering the value in the text field 'Cluster Name'. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

SMPP Encoding for GSM7/UCS2

Using CLI

You can set the 'SMPP Encoding for GSM7 (DCS=0)' and 'SMPP Encoding for UCS2 (DCS=8)' value by issuing the commands `smsc set smppencodingforgsm7` / `smsc set smppencodingforucs2` with appropriate parameters as described below. You can verify this by issuing the commands `smsc get smppencodingforgsm7` / `smsc get smppencodingforucs2` which will display the value set for these property. When GSM8 encoding type no recoding of message content is made.

```
Name
    smsc set smppencodingforgsm7
    smsc set smppencodingforucs2

SYNOPSIS
    smsc set smppencodingforgsm7 <UTF8|UNICODE|GSM7>
    smsc set smppencodingforucs2 <UTF8|UNICODE|GSM7>

DESCRIPTION
    These commands are used to set the Encoding Scheme at SMPP side for
    different GSM data coding schemas (DCS).

    For GSM7 encoding (DCS = 0) you must use the command
    smsc set smppencodingforgsm7, in order to set text encoding style.

    For UCS2 encoding (DCS = 8), you must use the command
    smsc set smppencodingforucs2, in order to set text encoding style.

    At the SMPP side, messages accept 3 different encoding schemes namely UTF8,
    UNICODE and GSM7 (8-bit), for both sending and receiving messages. The SMSC
    can be configured to accept one of them (the one that ESME supports). If this
    is not set, then the default encoding sheme is UTF8. For GSM8 encoding (DCS = 4),
    no charset encoding made in the SMSC.

EXAMPLES
    smsc set smppencodingforgsm7 utf8
    or
    smsc set smppencodingforucs2 unicode
```

Using GUI

Procedure: Set SMPP Encoding for GSM7 and UCS2 using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Properties' tab in the GUI.
3. You can specify the encoding scheme by choosing from the values (UTF8 | UNICODE | GSM7) in the list for 'SMPP Encoding for GSM7' (DCS=0) or 'SMPP Encoding for UCS2' (DCS=8). For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

Store And Forward Mode

Using CLI

You can set the 'Store And Forward Mode' value by issuing the command `smsc set storeandforwordmode` with appropriate parameters as described below. You can verify this by issuing the command `smsc get storeandforwordmode` which will display the value set for this property.

```
Name
    smsc set storeandforwordmode

SYNOPSIS
    smsc set storeandforwordmode <normal | fast>

DESCRIPTION
    This command is used to set the storeandforwordmode value.
    storeandforwordmode has two possible values:

    normal - StoreAndForward mode is used for incoming
    smpp StoreAndForward messages and all SS7 and SIP messages. All
    the incoming messages into SMSC will be persisted before trying
    for delivery.

    fast (default) - ForwardAndStore mode is used for incoming smpp
    StoreAndForward messages and all SS7 and SIP messages. This option
    can be switched without SMSC restart. All the incoming messages into
    SMSC will be tried for delivery first and only if delivery fails,
    it will be persisted for later re-try.

    Datagramm and Transactional modes will work in the same way for both normal and
fast modes.

EXAMPLES
    smsc get storeandforwordmode fast
```

Using GUI

Procedure: Set Store And Forward Mode using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Properties' tab in the GUI.
3. You can specify the Store And Forward Mode by selecting the value from the dropdown field 'Store And Forward Mode'. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

National language locking shift table

Using CLI

You can set the 'National language locking shift table' value by issuing the command smsc set nationallanguagelockingshift with appropriate parameters as described below. You can verify this by issuing the command smsc get nationallanguagelockingshift which will display the value set for this property.

```
Name
    smsc set nationallanguagelockingshift

SYNOPSIS
    smsc set nationallanguagelockingshift <NationalLanguageIdentifier>

DESCRIPTION
    National language locking shift table can be configured for messages
    that have come via SMPP, do not have UDHs inside and have GSM7 encoding
    (DCS==0).
    The default GSM data coding table is mostly used. Possible values:
    = 0: default GSM data coding table
    = 13: urdu (arabic) national language shift table
    This value can be also configured at ESME level.

EXAMPLES
    smsc set nationallanguagelockingshift 0
    smsc set nationallanguagelockingshift 13
```

Using GUI

Procedure: Set National language locking shift table using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Properties' tab in the GUI.
3. You can specify the National language locking shift table by entering the value in the text field 'National language locking shift'. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

National language single shift table

Using CLI

You can set the 'National language single shift table' value by issuing the command smsc set nationallanguagesingleshift with appropriate parameters as described below. You can verify this by

issuing the command smsc get nationallanguagesingleshift which will display the value set for this property.

```
Name
    smsc set nationallanguagesingleshift

SYNOPSIS
    smsc set nationallanguagesingleshift <NationalLanguageIdentifier>

DESCRIPTION
    National language single shift table can be configured for messages
    that have come via SMPP, do not have UDHs inside and have GSM7 encoding
    (DCS==0).
    The default GSM data coding table is mostly used. Possible values:
    = 0: default GSM data coding table
    = 13: urdu (arabic) national language shift table
    This value can be also configured at ESME level.

EXAMPLES
    smsc set nationallanguagesingleshift 0
    smsc set nationallanguagesingleshift 13
```

Using GUI

Procedure: Set National language single shift table using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Properties' tab in the GUI.
3. You can specify the National language single shift table by entering the value in the text field 'National language single shift'. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

Subscriber busy due delay (in sec)

Using CLI

You can set the 'Subscriber busy due delay (in sec)' value by issuing the command `smsc set subscriberbusyduedelay` with appropriate parameters as described below. You can verify this by issuing the command `smsc get subscriberbusyduedelay` which will display the value set for this property.

```
Name
    smsc set subscriberbusyduedelay

SYNOPSIS
    smsc set subscriberbusyduedelay <subscriber-busy-due-delay>

DESCRIPTION
    This command is used to set a value for subscriber-busy-due-delay (in seconds).
    This parameter specifies the delay time period in seconds when there has
    been a delivery failure with the cause 'subscriber busy'.

EXAMPLES
    smsc set subscriberbusyduedelay 2
```

Using GUI

Procedure: Set Subscriber busy due delay (in sec) using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Properties' tab in the GUI.
3. You can specify the Subscriber busy due delay by entering the value in the text field 'Subscriber busy due delay (in sec)'. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

First due delay (in sec)

Using CLI

You can set the 'First due delay (in sec)' value by issuing the command `smsc set firstduedelay` with appropriate parameters as described below. You can verify this by issuing the command `smsc get firstduedelay` which will display the value set for this property.

```
Name
    smsc set firstduedelay

SYNOPSIS
    smsc set firstduedelay <first-due-delay>

DESCRIPTION
    This command is used to set a value for first-due-delay (in seconds). This
    parameter specifies the delay time period in seconds between message incoming time
    and first delivery attempt.

EXAMPLES
    smsc set firstduedelay 60
```

Using GUI

Procedure: Set First due delay (in sec) using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Properties' tab in the GUI.
3. You can specify the First due delay by entering the value in the text field 'First due delay (in sec)'. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

Second due delay (in sec)

Using CLI

You can set the 'Second due delay (in sec)' value by issuing the command `smsc set secondduedelay` with appropriate parameters as described below. You can verify this by issuing the command `smsc get secondduedelay` which will display the value set for this property.

```
Name
    smsc set secondduedelay

SYNOPSIS
    smsc set secondduedelay <second-due-delay>

DESCRIPTION
    This command is used to set a value for second-due-delay (in seconds). This
    parameter specifies the delay time period in seconds between the first and second
    delivery attempt (i.e. if the first delivery attempt failed).

EXAMPLES
    smsc set secondduedelay 5
```

Using GUI

Procedure: Set Second due delay (in sec) using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Properties' tab in the GUI.
3. You can specify the Second due delay by entering the value in the text field 'Second due delay (in sec)'. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

Max due delay (in sec)

Using CLI

You can set the 'Max due delay (in sec)' value by issuing the command `smsc set maxduedelay` with appropriate parameters as described below. You can verify this by issuing the command `smsc get maxduedelay` which will display the value set for this property.

```
Name
    smsc set maxduedelay

SYNOPSIS
    smsc set maxduedelay <maxduedelay>

DESCRIPTION
    This command is used to set a value for max-due-delay (in seconds). This parameter
    specifies the maximum possible delay time period in seconds between delivery
    attempts.

EXAMPLES
    smsc set maxduedelay 3600
```

Using GUI

Procedure: Set Max due delay (in sec) using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Properties' tab in the GUI.
3. You can specify the Max due delay by entering the value in the text field 'Max due delay (in sec)'. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

Due delay multiplicator

Using CLI

You can set the 'Due delay multiplicator' value by issuing the command `smsc set duedelaymultiplicator` with appropriate parameters as described below. You can verify this by issuing the command `smsc get duedelaymultiplicator` which will display the value set for this property.

```
Name
    smsc set duedelaymultiplicator

SYNOPSIS
    smsc set duedelaymultiplicator <due-delay-multiplicator>

DESCRIPTION
    This command is used to set a value for due-delay-multiplicator. This parameter
    specifies the delay multiplicator value before another delivery attempt (after
    failure) is made.

    After a message delivery failure (if message validity period is not over and the
    failure is temporary), a delay period is induced before the next delivery attempt.
    This delay period is calculated as follows:

    Delay after the first delivery failure =
                second-due-delay

    Delay after every consecutive delivery failure =
                prev-due-delay * due-delay-multiplicator / 100
                    where prev-due-delay is the delay at the previous step.

EXAMPLES
    smsc set duedelaymultiplicator 200

    Lets take an example where the First due delay is 60 seconds, Second due delay is
    300 seconds, and the due-delay-multiplicator is 200, the attempts will be made
    as below:

    First attempt will be after 60 seconds (1 min)
    [delay is configured in First due delay]

    Second attempt will be after 300 seconds (5 min)
    [delay is configured in Second due delay assuming
    delivery failed not because of "Subscriber busy"]

    Third attempt will be after 600 sec (10 min)
    [delay is calculated based on Due delay multiplicator]
    Delay = 300 * 200 / 100 = 600

    Fourth attempt will be after 1200 sec (20 min)
    [delay is calculated based on Due delay multiplicator]
    Delay = 600 * 200 / 100 = 1200

    Fifth attempt will be after 2400 sec (40 min)
    [delay is calculated based on Due delay multiplicator]
    Delay = 1200 * 200 / 100 = 2400
```

Using GUI

Procedure: Set Due delay multiplicator using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Properties' tab in the GUI.
3. You can specify the Due delay multiplicator by entering the value in the text field 'Due delay multiplicator'. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

6.1.2. SS7 Settings

SMSC Global Title

Using CLI

You can set the 'SMSC Gobal Title' by issuing the command `smsc set scgt` with appropriate parameters as described below. You can verify this by issuing the command `smsc get scgt` which will display the value set for this property.

```
Name
        smsc set scgt

SYNOPSIS
        smsc set scgt <globalTitle> networkid <networkId>

DESCRIPTION
        This command is used to set a value for SMSC Global Title.

        networkId - a specifies Global Title for a virtual SS7
        subnetwork (this is for Multi-tenancy support). By using of
        this command with different networkIds you can specify
        Global Titles for several subnetworks.
        If this parameter is skipped - networkId will be set to "0"
        when Global Title creation (master networkId).
        When we do not specify Global Title for some networkid -
        Global Title for master networkId will be used. When we
        use "0" as Global Title value
        (like "smsc set scgt 0 networkid <xxx>") -
        this will just clear Global Title for an specified networkid.
```

Using GUI

Procedure: Set SMSC Global Title using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'SS7 Settings' tab in the GUI.
3. You can specify the SMSC Global Title by entering values into fields pair 'SMSC Gateway Global Title Indicator Network Id' and 'SMSC Gateway Global Title'. You are able to set Global Title for definite networkId. Setting of Global Title for networkId to "0" leads clearing of Global Title for networkId. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

SCCP Global Title type

Using CLI

You can set the 'SCCP Global Title type' by issuing the command `smsc set gti` with appropriate parameters as described below. You can verify this by issuing the command `smsc get gti` which will display the value set for this property.

```
Name
        smsc set gti

SYNOPSIS
        smsc set gti 0001|0010|0011|0100

DESCRIPTION
        This command is used to set the value of SCCP Global Title type.
        This Global Title type will be used for SCCP outgoing messages.
        Default value for ITU-T is 0100.
        Global title 0001 - Nature of address indicator included
        Global title 0010 - Translation type included
        Global title 0011 - Translation type, Numbering plan and Encoding
                            scheme included
        Global title 0100 - Translation type, Numbering plan, Encoding
                            scheme and Nature of address indicator included
```

Using GUI

Procedure: Set SCCP Global Title type using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'SS7 Settings' tab in the GUI.
3. You can specify the SCCP Global Title type by entering the value in the text field 'SCCP Global

Title'. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.

4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

SCCP translation type value

Using CLI

You can set the 'SCCP translation type value' by issuing the command `smsc set tt` with appropriate parameters as described below. You can verify this by issuing the command `smsc get tt` which will display the value set for this property.

```
Name
        smsc set tt

SYNOPSIS
        smsc set tt <translation type value>

DESCRIPTION
        This command is used to set the value of SCCP translation type
        value. Translation type value will be used for SCCP outgoing
        messages. Default value for ITU-T is 0.
```

Using GUI

Procedure: Set SCCP translation type value using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'SS7 Settings' tab in the GUI.
3. You can specify the SCCP translation type value by entering the value in the text field 'SCCP translation type'. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

SMSC Sub System Number (SSN)

Using CLI

You can set the 'SMSC SSN' by issuing the command `smsc set scssn` with appropriate parameters as described below. You can verify this by issuing the command `smsc get scssn` which will display the value set for this property.

```
Name
        smsc set scssn

SYNOPSIS
        smsc set scssn <smscSubSystemNumber>

DESCRIPTION
        This command is used to set the value of SMSC Sub System Number (SSN). Issuing
        this command in CLI will set the SSN value but you must ensure that the SSN
number
        is properly configured in the TCAP Stack in the xml descriptor file
        "mobicents-smscgateway-version/jboss-5.1.0.GA/server/default/deploy/
        mobicents-smsc-server/META-INF/jboss-beans.xml".
```

Using GUI

Procedure: Set SMSC Sub System Number (SSN) using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'SS7 Settings' tab in the GUI.
3. You can specify the SMSC Sub System Number (SSN) by entering the value in the text field 'SMSC Gateway subsystem number'. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

HLR Sub System Number (HLR SSN)

Using CLI

You can set the 'HLR SSN' by issuing the command `smsc set hlrssn` with appropriate parameters as described below. You can verify this by issuing the command `smsc get hlrssn` which will display the value set for this property.

```
Name
        smsc set hlrssn

SYNOPSIS
        smsc set hlrssn <hlrSubSystemNumber>

DESCRIPTION
        This command is used to set the value of HLR Sub System Number (SSN).
```

Using GUI

Procedure: Set HLR Sub System Number (SSN) using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'SS7 Settings' tab in the GUI.
3. You can specify the HLR Sub System Number (SSN) by entering the value in the text field 'HLR subsystem number'. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

MSC Sub System Number (SSN)

Using CLI

You can set the 'MSC SSN' by issuing the command `smsc set mscssn` with appropriate parameters as described below. You can verify this by issuing the command `smsc get mscssn` which will display the value set for this property.

```
Name
        smsc set mscssn

SYNOPSIS
        smsc set mscssn <mscSubSystemNumber>

DESCRIPTION
        This command is used to set the value of MSC Sub System Number (SSN).
```

Using GUI

Procedure: Set MSC Sub System Number (SSN) using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'SS7 Settings' tab in the GUI.
3. You can specify the MSC Sub System Number (SSN) by entering the value in the text field 'MSC subsystem number'. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

MAP Application Context version

Using CLI

You can set the 'MAP Application Context version' by issuing the command `smsc set maxmapv` with appropriate parameters as described below. You can verify this by issuing the command `smsc get maxmapv` which will display the value set for this property.

```
Name
        smsc set maxmapv

SYNOPSIS
        smsc set maxmapv <version-number>

DESCRIPTION
        This command is used to set the value of MAP Application Context version. The
        version number set here will be used for SMS messages exchanged. RestComm SMSC
        Gateway supports version negotiation. So if you set this to a higher version
(say
        for example version 3, however your network only understands version 2), the
SMSC Gateway
        will automatically do the version negotiation and exchange V2 messages when V3
        exchange fails. However this causes additional messages to be exchanged and
        increases the overall load on the system. Therefore it is advisable to always
set
        the correct version.
```

Using GUI

Procedure: Set MAP Application Context version using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'SS7 Settings' tab in the GUI.
3. You can specify the MAP Application Context version by entering the value in the text field 'MAP version supported'. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

Max Message Length Reducer

Using CLI

You can set the 'Max Message Length Reducer' value by issuing the command `smsc set maxmessagelengthreducer` with appropriate parameters as described below. You can verify this by issuing the command `smsc get maxmessagelengthreducer` which will display the value set for this

property.

```
Name
    smsc set maxmessagelengthreducer

SYNOPSIS
    smsc set maxmessagelengthreducer <max-message-length-reducer>

DESCRIPTION
    This command is used to set an integer value for max-message-length-reducer.
    The recommended value is 6. Possible values are numbers from 0 to 12.

    Empty TC-BEGIN will be used if the message length is greater than the maximum
    possible message length minus the value specified for max-message-length-reducer.

    (message-length > max-possible-message-length - max-message-length-reducer)

    Empty TC-BEGIN is used in MAP Version 2 and 3 for forwardSM and Mt-ForwardSM
    requests. In MAP Version 2 the dailog portion
    (ApplicationContextName, MAPOpenInfo primitive) and the component portion
    (forwardSM and mt-ForwardSM requests) may both together be too long to fit within
    a MTP message. In Empty TC-BEGIN case, it first sends the dailog portion in
    TC-BEGIN followed by the component portion in the next TC-CONTINUE.
    Whether empty TC-BEGIN is used or not depends on the length of a message and the
    length of SCCP addresses. This option increases the guarantee of delivery of a
    message  to some network.

EXAMPLES
    smsc set maxmessagelengthreducer 6
```

Using GUI

Procedure: Set Max Message Length Reducer using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'SS7 Settings' tab in the GUI.
3. You can specify the Max Message Length Reducer by entering the value in the text field 'Max Message Length Reducer'. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

Pre-configured HLR address for SRI messages

Using CLI

You can set a pre-configured HLR address for SRI messages by issuing the command `smsc set hrhlrnumber` with appropriate parameters as described below. You can remove this pre-configured address by issuing the command `smsc remove hrhlrnumber`. You can verify this by issuing the command `smsc get hrhlrnumber` which will display the value set for this property.

```
Name
    smsc set hrhlrnumber
    smsc remove hrhlrnumber

SYNOPSIS
    smsc set hrhlrnumber <hlr GT digits> networkid <networkId>
    smsc remove hrhlrnumber networkid <networkId>

DESCRIPTION
    This command is used to set a pre-configured HLR address for SRI messages.

    In some scenarios it may be required to set a HLR address instead of a
    MSISDN address into the SCCP 'CalledPartyAddress' of 'SendRoutingInfo'
    requests issued by the SMSC GW in both mobile terminated and home routing
    modes. In such cases, you must set this parameter 'hrhlrnumber' to a
    pre-configured HLR address.

    For all other scenarios where this is not required, you may leave this
    parameter empty. When this is empty, the SCCP 'CalledPartyAddress' of
    'SendRoutingInfo' request will be set to the destination MSISDN of a
    subscriber.

    networkId - specifies a virtual SS7 subnetwork (this is for Multi-tenancy
    support). By using of this command with different networkIds you can
    specify hrhlrnumber for several subnetworks.
    If this parameter is skipped - networkId will be set to "0".
```

Using GUI

Procedure: Set Pre-configured HLR address for SRI messages using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'SS7 Settings' tab in the GUI.
3. You can specify (set or remove) the Pre-configured HLR address for SRI messages by entering appropriate values for a specified networkID. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

SRI responses Cache Time (in secs)

Using CLI

You can set a SRI responses Cache Time by issuing the command `smsc set sriresponselivetime` with appropriate parameters as described below. You can verify this by issuing the command `smsc get sriresponselivetime` which will display the value set for this property.

```
Name
    smsc set sriresponselivetime

SYNOPSIS
    smsc set sriresponselivetime <time in seconds>

DESCRIPTION
    This command is used to set a SRI responses Cache Time.

    SMSC GW can store successfull SendRoutinInfo (SRI) responses (with IMSI and
    NetworkNodeNumber data) into an internal cach for some configurable time.
    SMSC GW parameter "sriresponselivetime" specifies the minimum time value
    for storing of a response. Caching of SRI responses takes some system
    resources and is recommended only if you need it for some scenarious (like
    you send firstly only an SRI request and do not deliver a message just to
    understend IMSI / NetworkNodeNumber (this scenario is achieavable by mproc
    rules) and then send a message in short time if needed).

DEFAULT VALUE
    0 - this means no caching.

EXAMPLES
    smsc set sriresponselivetime 0
```

Using GUI

Procedure: Set SRI responses Cache Time using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'SS7 Settings' tab in the GUI.
3. You can specify (set or remove) the "SRI responses Cache Time (in secs)" by entering appropriate values. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

6.1.3. Cassandra Settings

You can manage Cassandra Settings using the CLI or GUI. Note that the modified settings will become effective only when the SMSC is re-started.

Cassandra Configuration - Host Addresses

Using CLI

You can set the 'host addresses' value for Cassandra settings by issuing the command `smsc set dbhosts` with appropriate parameters as described below. You can verify this by issuing the command `smsc get dbhosts` which will display the value set for this property.

```
Name
    smsc set dbhosts

SYNOPSIS
    smsc set dbhosts <host-ip>

DESCRIPTION
    This command is used to set the host-ip addresses for Cassandra Database access.

EXAMPLES
    smsc set dbhosts 127.0.0.1
```

Using GUI

Procedure: Set Cassandra Configuration - Host Addresses using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Cassandra' tab in the GUI.
3. You can specify the host-ip address by entering appropriate values in the text field 'Host Address'. For more details of these parameters, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

Cassandra Configuration - Port

Using CLI

You can set the 'port' value for Cassandra settings by issuing the command `smsc set dbport` with appropriate parameters as described below. You can verify this by issuing the command `smsc get dbport` which will display the value set for this property.

```
Name
    smsc set dbport

SYNOPSIS
    smsc set dbport <port>

DESCRIPTION
    This command is used to set the host-ip address for Cassandra Database
    access. Pass comma separated values if Cassandra is setup in cluster
    and can be accessed via multiple IP's

EXAMPLES
    smsc set dbport 9042
```

Using GUI

Procedure: Set Cassandra Configuration - Port using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Cassandra' tab in the GUI.
3. You can specify the host-ip address and port by entering appropriate values in the text field 'Port'. For more details of these parameters, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

Cassandra Configuration - Keyspace Name

Using CLI

You can set the 'DB Keyspace Name' by issuing the command `smsc set keyspacename` with appropriate parameters as described below. You can verify this by issuing the command `smsc get keyspacename` which will display the value set for this property.

```
Name
    smsc set keyspacename

SYNOPSIS
    smsc set keyspacename <keyspacename>

DESCRIPTION
    This command is used to set the Keyspace name for Cassandra Database. If you use
    the script available in the distributive the name will be set to 'RestCommSMSC' by
    default.
```

Using GUI

Procedure: Set Cassandra Configuration - Keyspace Name using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Cassandra' tab in the GUI.
3. You can specify the Keyspace Name by entering the value in the text field 'Keyspace Name'. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

Cassandra Configuration - Cluster Name

Using CLI

You can set the 'DB Cluster Name' value by issuing the command `smsc set clustername` with appropriate parameters as described below. You can verify this by issuing the command `smsc get clustername` which will display the value set for this property.

```
Name
    smsc set clustername

SYNOPSIS
    smsc set clustername <cluster-name>

DESCRIPTION
    This command is used to set the Cluster name for Cassandra Database. If you use
    the script available in the distributive the name will be set to 'RestCommSMSC' by
    default.

EXAMPLES
    smsc set clustername RestCommSMSC
```

Using GUI

Procedure: Set Cassandra Configuration - Cluster Name using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Cassandra' tab in the GUI.
3. You can specify the Cluster Name by entering the value in the text field 'Cluster Name'. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.

4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

Cassandra Configuration - Removing Live Tables Days

Using CLI

You can set the 'Removing Live Tables Days' value by issuing the command `smsc set removinglivetabledays` with appropriate parameters as described below. You can verify this by issuing the command `smsc get removinglivetabledays` which will display the value set for this property.

```
Name
    smsc set removinglivetablesdays

SYNOPSIS
    smsc set removinglivetablesdays <value>

DESCRIPTION
    This command is used to configure the SMSC to automatically drop LIVE tables from
    the Cassandra Database. The SMSC will attempt to delete tables just after
    midnight and after every SMSC restart.

PARAMETERS
    removinglivetablesdays      - This parameter is used to specify the number
                      of days the LIVE tables should be kept before
                      attempting to drop them automatically.

                      If this value is specified as "0", the SMSC
                      will not drop tables automatically. In this case
                      you must manually drop tables.

                      You must specify a value of 3 or more. You can
                      not set this value to 1 or 2 days. This is to
                      ensure the tables will be kept for a minimum
                      of 2 days after creation date.

                      The SMSC wil attempt to delete tables for one
                      day. If the Cassandra Database keeps tables for
                      older days, then the administrator should drop
                      these manually.
```

Using GUI

Procedure: Set Cassandra Configuration - Removing Live Tables Days

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing.

Switch to the 'Cassandra' tab in the GUI.

3. You can specify the Removing Live Tables Days by entering the value in the text field 'Removing Live Tables Days'. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

Cassandra Configuration - Removing Live Tables Days

Using CLI

You can set the 'Removing Live Tables Days' value by issuing the command `smsc set removingarchivetabledays` with appropriate parameters as described below. You can verify this by issuing the command `smsc get removingarchivetabledays` which will display the value set for this property.

```
Name
    smsc set removingarchivetabledays

SYNOPSIS
    smsc set removingarchivetabledays <value>

DESCRIPTION
    This command is used to configure the SMSC to automatically drop ARCHIVE tables
    from the Cassandra Database. The SMSC will attempt to delete tables just after
    midnight and after every SMSC restart.

PARAMETERS
    removingarchivetabledays    - This parameter is used to specify the number
                        of days the ARCHIVE tables should be kept before
                        attempting to drop them automatically.

                        If this value is specified as "0", the SMSC
                        will not drop tables automatically. In this case
                        you must manually drop tables.

                        You must specify a value of 3 or more. You can
                        not set this value to 1 or 2 days. This is to
                        ensure the tables will be kept for a minimum
                        of 2 days after creation date.

                        The SMSC wil attempt to delete tables for one
                        day. If the Cassandra Database keeps tables for
                        older days, then the administrator should drop
                        these manually.
```

Using GUI

Procedure: Set Cassandra Configuration - Removing Archive Tables Days

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Cassandra' tab in the GUI.
3. You can specify the Removing Archive Tables Days by entering the value in the text field 'Removing Archive Tables Days'. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

6.1.4. Scheduler Settings

You can modify Scheduler settings using the CLI or GUI. The modified settings for Fetch Period will become effective only when the SMSC is re-started. However modified settings for Max Rows and Max Activity Count will take effect immediately.

Fetch Period (in ms)

Using CLI

You can set the 'Fetch Period' value by issuing the command `smsc set fetchperiod` with appropriate parameters as described below. You can verify this by issuing the command `smsc get fetchperiod` which will display the value set for this property.

```
Name
    smsc set fetchperiod

SYNOPSIS
    smsc set fetchperiod <fetch-period>

DESCRIPTION
    This command is used to set the fetch period value in milli-seconds for the
    Cassandra database. The parameter fetch-period specifies the time
    period (in milli-seconds) of fetching messages for delivery from the database.
    The default value is 5 seconds.

EXAMPLES
    smsc set fetchperiod 5000
```

Using GUI

Procedure: Set Cassandra Configuration - Cluster Name using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.

2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Scheduler' tab in the GUI.
3. You can specify the Fetch Period by entering the value in the text field 'Fetch Period (in ms)'. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

Max Rows

Using CLI

You can set the 'Max Rows' value by issuing the command `smsc set fetchmaxrows` with appropriate parameters as described below. You can verify this by issuing the command `smsc get fetchmaxrows` which will display the value set for this property.

```
Name
    smsc set fetchmaxrows

SYNOPSIS
    smsc set fetchmaxrows <fetch-max-rows>

DESCRIPTION
    This command is used to set the maximum message fetching count for every fetching
    step from the database.
    The default value is 100 messages.

EXAMPLES
    smsc set fetchmaxrows 200
```

Using GUI

Procedure: Set Max Rows using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Scheduler' tab in the GUI.
3. You can specify the Max Rows by entering the value in the text field 'Max Rows'. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

Max Activity Count

Using CLI

You can set the 'Max Activity Count' value by issuing the command `smsc set maxactivitycount` with appropriate parameters as described below. You can verify this by issuing the command `smsc get maxactivitycount` which will display the value set for this property.

```
Name
    smsc set maxactivitycount

SYNOPSIS
    smsc set maxactivitycount <max-activity-count>

DESCRIPTION
    This command is used to set the maximum count of delivering activities that are
    possible at the same time. 'Count of delivering activities' means the count of
    messages that are in the state 'delivering' (messages that are fetched from the
    database and may be already sent or are going to be sent but
    no delivery acception/rejection has been received).
    When the delivery process of a message is in progress, field LIVE.IN_SYSTEM==2.

EXAMPLES
    smsc set maxactivitycount 500
```

Using GUI

Procedure: Set Max Activity Count using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Scheduler' tab in the GUI.
3. You can specify the Max Activity Count by entering the value in the text field 'Max Activity Count'. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

Revise period after SMSC restart

Using CLI

You can set the 'Revise period (in seconds) after SMSC restart' value by issuing the command `smsc set revisesecondsonsmscstart` with appropriate parameters as described below. You can verify this by issuing the command `smsc get revisesecondsonsmscstart` which will display the value set for this property. If unspecified, the default value for this parameter is 60 seconds.

```
Name
    smsc set revisesecondsonsmscstart <seconds>

SYNOPSIS
    smsc set revisesecondsonsmscstart <seconds>

DESCRIPTION
    This command is used to set the revise period (in seconds). After every restart,
    the SMSC Gateway will revise the last 'x' seconds before shutdown to ensure that
    all the arrived messages are processed; where 'x' is the value set in seconds
    for the parameter 'revisesecondsonsmscstart' using this command.
EXAMPLES
    smsc set revisesecondsonsmscstart 30
```

Using GUI

Procedure: Set Revise Period using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Scheduler' tab in the GUI.
3. You can specify the revise period in seconds by entering the value in the text field 'Revise period after SMSC restart (sec)'. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

Cache timeout period

Using CLI

You can set the 'Cache timeout period' value by issuing the command `smsc set processingsmssettimeout` with appropriate parameters as described below. You can verify this by issuing the command `smsc get processingsmssettimeout` which will display the value set for this property. If unspecified, the default value for this parameter is 600 seconds. Generally, you may not have to modify this value.

```
Name
    smsc set processingsmssettimeout <seconds>

SYNOPSIS
    smsc set processingsmssettimeout <seconds>

DESCRIPTION
    This command is used to set the Cache timeout period (in seconds).
    Messages are cached in the SMSC until the processing is completed. In case of a
    delivery failure, these cached messages are force cleaned by the SMSC after
    waiting for the timeout period set for the parameter 'processingsmssettimeout'
    using this command.
EXAMPLES
    smsc set processingsmssettimeout 45
```

Using GUI

Procedure: Set Cache timeout Period using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Scheduler' tab in the GUI.
3. You can specify the Cache timeout period in seconds by entering the value in the text field 'Processing Sms set cache timeout (sec)'. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

Skipping of scheduled for the past and not yet sent messages ("In process" due_slot shifting).

Using CLI

This command will skip of processing / fetching of messages that was scheduled for delivering for the time in the past but have not yet delivered by SMSC GW because of SMSC GW was turned off or overloaded. <time in seconds> means the point of the time (actual current time - <time in seconds>) to which the point of processing / fetching of messages will be shifted. If the value "0" is provided this means SMSC GW will be shifted into an actual (current) time ("in process" due_slot will be shifted to "in time" due_slot). If the value is positive this means SMSC GW "in process" due_slot will be shifted into a some time in the past (for example if the value is 3600 - to the time before the current time ("in time" due_slot) for 1 hour). Negative values are not accepted. "In process" due_slot can be shifted only forward. It is not possible to shift "in process" due_slot backward (and resend messages that was already sent once).

```
Name
    smsc skipunsentmessages

SYNOPSIS
    smsc skipunsentmessages <time in seconds>

DESCRIPTION
    Executing of this command leads SMSC GW to switch "In process" due_slot
    forward to the current time or to some time before the current time.
    This is possible only if there is some lag in message processing by
    SMSC GW. This also leads of skipping of sending messages that were
    scheduled for time in the past but have not delivered so far.

EXAMPLES
    smsc skipunsentmessages 0
```

Using GUI

Procedure: Skipping of scheduled for the past and not yet sent messages ("In process" due_slot shifting) by the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Scheduler' tab in the GUI.
3. Set "Skip Unsent Messages (in sec) " field to 0 or positive value.
4. You must click on the button 'Save' that is below to skip scheduled messages. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

6.1.5. Diameter Settings

You can modify Diameter settings using the CLI or GUI.

Destination Realm

Using CLI

You can set the 'Destination Realm' value by issuing the command `smsc set diameterdestrealm` with appropriate parameters as described below. You can verify this by issuing the command `smsc get diameterdestrealm` which will display the value set for this property.

```
Name
    smsc set diameterdestrealm

SYNOPSIS
    smsc set diameterdestrealm <value>

DESCRIPTION
    This command is used to set the Diameter Destination Realm for connection to OCS.
    Default value is "mobicents.org".

EXAMPLES
    smsc set diameterdestrealm mobicents.org
```

Using GUI

Procedure: Set Diameter Destination Realm using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Diameter' tab in the GUI.
3. You can specify the Destination Realm by entering the value in the corresponding text field. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

Destination Host

Using CLI

You can set the 'Destination Host' value by issuing the command `smsc set diameterdesthost` with appropriate parameters as described below. You can verify this by issuing the command `smsc get diameterdesthost` which will display the value set for this property.

```
Name
    smsc set diameterdesthost

SYNOPSIS
    smsc set diameterdesthost <value>

DESCRIPTION
    This command is used to set the Diameter Destination Host for connection to OCS.
    Default value is "127.0.0.1".

EXAMPLES
    smsc set diameterdesthost 127.0.0.1
```

Using GUI

Procedure: Set Diameter Host using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Diameter' tab in the GUI.
3. You can specify the Destination Host by entering the value in the corresponding text field. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

Destination Port

Using CLI

You can set the 'Destination Port' value by issuing the command `smsc set diameterdestport` with appropriate parameters as described below. You can verify this by issuing the command `smsc get diameterdestport` which will display the value set for this property.

```
Name
    smsc set diameterdestport

SYNOPSIS
    smsc set diameterdestport <value>

DESCRIPTION
    This command is used to set the Diameter Destination Port for connection to OCS.
    Default value is 3868.

EXAMPLES
    smsc set diameterdestport 3868
```

Using GUI

Procedure: Set Diameter Port using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Diameter' tab in the GUI.
3. You can specify the Destination Port by entering the value in the corresponding text field. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

Destination Username

Using CLI

You can set the 'Destination Username' value by issuing the command `smsc set diameterusername` with appropriate parameters as described below. You can verify this by issuing the command `smsc get diameterusername` which will display the value set for this property.

```
Name
    smsc set diameterusername

SYNOPSIS
    smsc set diameterusername <value>

DESCRIPTION
    This command is used to set the Diameter Username for connection to OCS.

EXAMPLES
    smsc set diameterdestusername svinu
```

Using GUI

Procedure: Set Diameter Username using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Diameter' tab in the GUI.
3. You can specify the Username by entering the value in the corresponding text field. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

MO accepting and charging settings

Using CLI

You can set the 'MO (mobile originated) Charging' value by issuing the command `smsc set mocharging` with appropriate parameters as described below. You can verify this by issuing the command `smsc get mocharging` which will display the value set for this property.

```
Name
    smsc set mocharging

SYNOPSIS
    smsc set mocharging <accept|reject|diameter>

DESCRIPTION
    This command is used to set the value of the parameter 'moCharging' to
    an appropriate value. This value is set to "accept" by default.
    - accept - all Mobile Originated messages are accepted
    - reject - all Mobile Originated messages are rejected
    - diameter - all Mobile Originated messages are charged by OCS via
      Diameter, prior to being sent

EXAMPLES
    smsc set mocharging accept
```

Using GUI

Procedure: Set MO Charge using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Diameter' tab in the GUI.
3. You can set 'Mobile Originated SMS Charged' value to true or false, in the corresponding list. For

more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.

4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

ESME Charge Settings

Using CLI

You can set the 'ESME Originated SMS Charged' value by issuing the command `smsc set txsmppcharging` with appropriate parameters as described below. You can verify this by issuing the command `smsc get txsmppcharging` which will display the value set for this property.

```
Name
    smsc set txsmppcharging

SYNOPSIS
    smsc set txsmppcharging <none|selected|all>

DESCRIPTION
    This command is used to set the value of the parameter 'txsmppcharging' to none,
    selected or all.

    If this is set to 'all', all ESME Originated messages will be charged by OCS via
    Diameter, prior to being sent.

    If this is set to 'selected', only those messages originating from ESMEs marked
    with the parameter 'charging-enabled'=true at the time of ESME creation  will be
    charged by OCS via Diameter, prior to being sent.

    If this is set to 'none', none of the ESME Originated messages will be charged by
    OCS via Diameter, prior to being sent.

    The parameter 'txsmppcharging' is set to 'none' by default.

EXAMPLES
    smsc set txsmppcharging selected
```

Using GUI

Procedure: Set ESME Charge using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Diameter' tab in the GUI.
3. You can set 'ESME Originated SMS Charged' value to none, selected or all, in the corresponding list. For more details of this parameter, please refer to the description of the CLI command for

the same in the preceding section.

4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

SIP Charge Settings

Using CLI

You can set the 'SIP Originated SMS Charged' value by issuing the command `smsc set txsipcharging` with appropriate parameters as described below. You can verify this by issuing the command `smsc get txsipcharging` which will display the value set for this property.

```
Name
    smsc set txsipcharging

SYNOPSIS
    smsc set txsipcharging <none|selected|all>

DESCRIPTION
    This command is used to set the value of the parameter 'txsipcharging' to none,
    selected or all.

    If this is set to 'all', all SIP Originated messages will be charged by OCS via
    Diameter, prior to being sent.

    If this is set to 'selected', only those messages originating from SIPs marked
    with the parameter 'charging-enabled'=true at the time of SIP creation  will be
    charged by OCS via Diameter, prior to being sent.

    If this is set to 'none', none of the SIP Originated messages will be charged by
    OCS via Diameter, prior to being sent.

    The parameter 'txsipcharging' is set to 'none' by default.

EXAMPLES
    smsc set txsipcharging selected
```

Using GUI

Procedure: Set SIP Charge using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Diameter' tab in the GUI.
3. You can set 'SIP Originated SMS Charged' value to none, selected or all, in the corresponding list. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.

4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

Home routing Charge Settings

Using CLI

You can set the 'Home routing Originated SMS Charged' value by issuing the command `smsc set hrcharging` with appropriate parameters as described below. You can verify this by issuing the command `smsc get hrcharging` which will display the value set for this property.

```
Name
    smsc set hrcharging

SYNOPSIS
    smsc set hrcharging <accept|reject|diameter>

DESCRIPTION
    This command is used to set the value of the parameter 'hrcharging' to
    an appropriate value. This value is set to "accept" by default. This
    option works like mocharging option but affects on SS7 messages in home
    routing mode from upper SMSC (mocharging affects on mobile originated
    SS7 incoming messages).
    - accept - all Home Routing originated messages are accepted
    - reject - all Home Routing originated messages are rejected
    - diameter - all Home Routing Originated messages are charged by OCS via
      Diameter, prior to being sent

EXAMPLES
    smsc set hrcharging accept
```

Using GUI

Procedure: Set Home Routing Charge using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Diameter' tab in the GUI.
3. You can set 'Home Routed SMS Charged' value to none, selected or all, in the corresponding list. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

6.1.6. Home Routing Settings

RestComm SMSC Gateway supports Non Transparent Home Routing as explained in 3GPP 23.840 5.2.3. This option can be disabled by setting "hrcharging" option to "reject" (see Home routing Charge Settings). SMSC GW will accept SendRoutinInfo requests from a remote SMSC, then create a unique "correlationId" value for a requested MSISDN, then send SendRoutinInfo request to HLR, store received by HLR info into a cache and sends back to the remote SMSC SendRoutinInfo response with "correlationId" value in IMSI field and SMSC GlobalTytle (or another special configured GlobalTytle value) in LocationInfo field. Then SMSC GW will accept MtForwardSM messages from the remote SMSC, seaches in the cache for "correlationId" and HLR subscriber's and use this data for processing a message further.

Correlation table CC and MCC-MNC for home routing mode managing.

For home routing mode we may need to fill a special table for correlation between CC (CountryCodes) of incoming MSISDN address and MCC-MNC prefix of a generated correlationId value. CorrelationId value will be returned as IMSI in SRI response to an upper SMSC GW. As an extra field "smsc" field can be specified for each entry of this table. If this field is specified and not empty then this value will be returned as a LocationInfo in SRI response to an upper SMSC GW, else SMSC GW address will be returned. Correlation table is stored into "jboss-5.1.0.GA/server/<server instanse, for example 'default'>"/data" and has name "SmscManagement_cc_mccmnc.xml". The content of this should follow the following template:

```
<?xml version="1.0" encoding="UTF-8" ?>
  <CcMccmncCollection>
  <ccMccmncList>
    <ccMccmnc countryCode="0111" mccMnc="77702"/>
    <ccMccmnc countryCode="0222" mccMnc="9999999"/>
    <ccMccmnc countryCode="02" mccMnc="8888888" smsc="06060606"/>
    <ccMccmnc countryCode="" mccMnc="22323"/>
  </ccMccmncList>
</CcMccmncCollection>
```

The file structure consists on one or several instances of countryCode / mccMnc pairs. When SMSC GW reads a correlation table from the file (or when a user add new entries into the table) it sorts it so that the longer CountryCode values (more detailed "CountryCode") are put at first places of the list then shorter CountryCode values. For example "44779" will be before "44". First found "mccMnc" value will be used for correlationId generating. When SMSC GW receives a SRI request from an upper SMSC GW it looks throught the table, checks if incoming MSISDN digits start from countryCode value from a table entry. The last entry in the correlation table must be empty ("") countryCode value. This entry will be used as a default value. All MSISDN that are not fit to any other entries will use mccMnc of the entry of "". You can also add an optional extra field "smsc" into any record. This value will be used as LocationInfo field in SendRoutinInfo response.

Living time of elements in correlation cache

Using CLI

In home routing mode SMSC GW specifies correlationId for any recieved SRI requests from upper SMSC. After it this correlationId value and corresponded to MSISDN data are stored into internal cache and will be used for processing of next coming MtForwardSM messages from other SMSC. Correlationidlivetime value defines how much time minimum is correlationId in cache. You can set the 'correlationidlivetime' option by issuing the command `smsc set correlationidlivetime` with appropriate parameters as described below. You can verify this by issuing the command `smsc get correlationidlivetime` which will display the value set for this property.

Also pay attention that when Telestax SMSC delivers messages received under "home routing" procedure SMSC GW will try to reuse location info and IMSI data that SMSC GW has obtained when request to HLR under "home routing" procedure. SMSC GW will try to reuse this info only till it is kept in correlationId cache. When ForwardAndStore and datagram modes this is usually achived in correlationidlivetime is 60 seconds. In StoreAndForward mode messages delivery can be started some time after messages have come to SMSC GW. This delay depends on both SMSC GW setting (first delivery attempt) and SMSC GW overloadload rate. For StoreAndForward mode we need to caluclate a proper correlationidlivetime value depending on othe SMSC setting. But do not make this value too big - this will waste memory and location info can occure too old.

```
Name
    smsc set correlationidlivetime

SYNOPSIS
    smsc set correlationidlivetime <digital option>

DESCRIPTION
    This command is used to set min time duration for which
    correlationId and corresponded data kept in cache. Max
    duratuion is two times more then correlationidlivetime.
    Value is in seconds.

    Default value: 60 (seconds).

EXAMPLES
    smsc set correlationidlivetime 90
```

Using GUI

Procedure: Setting of living time of elements in correlation cache using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Home Routing' tab in the GUI.
3. You can specify CDR generation option by selecting an appropriate value in the field "Correlation Id Cache Time (in secs)". For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.

4. You must click on the button 'Save' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

Bypassing of SRI request to a local HLR

Using CLI

SMSC GW home routing procedure for each networkId area can be configured to send or do not send SendRoutinInfo request to a local HLR. This is configured by parameter "hrsribypass". Default value is "false" - this means that SMSC GW will send SRI requests to local HLR before sending back SRI response.

Messageflow for home routing procedure when hrsribypass==false

SRI request: remote SMSC GW → Telscale SMSC GW

SRI request: Telscale SMSC GW → local HLR

SRI response: local HLR → Telscale SMSC GW

SRI response: Telscale SMSC GW → remote SMSC GW

MT request: remote SMSC GW → Telscale SMSC GW

MT response: Telscale SMSC GW → remote SMSC GW

Messageflow for home routing procedure when hrsribypass==true (SRI request to local HLR bypassing)

SRI request: remote SMSC GW → Telscale SMSC GW

SRI response: Telscale SMSC GW → remote SMSC GW

MT request: remote SMSC GW → Telscale SMSC GW

MT response: Telscale SMSC GW → remote SMSC GW You can set the 'hrsribypass' option by issuing the command `smsc set hrsribypass` with appropriate parameters as described below. You can verify this by issuing the command `smsc get hrsribypass` which will display the value set for this property.

```
Name
    smsc set hrsribypass

SYNOPSIS
    smsc set hrsribypass <digital option> networkid <networkId>

DESCRIPTION
    This command is used to set if SMSC GW will
    bypass a SRI to a local HLR.

    networkId - specifies a virtual SS7 subnetwork (this is for Multi-tenancy
    support). By using of this command with different networkIds you can
    specify hrsribypass for several subnetworks.
    If this parameter is skipped - networkId will be set to "0".
    If you have not specified hrsribypass parameter for a networkId
    then a master hrsribypass will be used (that was specified for
    networkId 0).

    Default value: false.

EXAMPLES
    smsc set hrsribypass true
    smsc set hrsribypass false networkid <networkId>
```

Using GUI

Procedure: Setting of living time of elements in correlation cache using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Home Routing' tab in the GUI.
3. You can specify hrsribypass option by selecting an appropriate value in the field "Bypassing of SRI request to a local HLR" for a specified networkId. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Save' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

Create an entry for correlation table CC and MCC-MNC

Using CLI

In home routing mode SMSC GW we may need to specify the correlation table CC and MCC-MNC (Correlation table CC and MCC-MNC for home routing mode managing.). You can add an entry of this table by issuing the command `smsc hrccmccmnc add` with appropriate parameters as described below.

```
Name
    smsc hrccmccmnc add

SYNOPSIS
    smsc hrccmccmnc add <countrycode> <mccmnc> smscgt <smsc-gt>

DESCRIPTION
    This command is used to add an entry to the correlation
    table CC and MCC-MNC.
    mscgt parameter is optional. If it is missed will be set]
    to "null" value. For "null" value for <smsc-gt> we can
    to specify "-1" value in CLI.

EXAMPLES
    smsc hrccmccmnc add 2223 55322 smscgt 733211232342
```

Using GUI

Procedure: Creation an entry for correlation table CC and MCC-MNC using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Home Routing CC - MCC MNC Table' in the left panel.
2. You can add an entry of the table - you can specify "Country Code", "Mobile Country Code and Mobile Network Code", "Global Title" (optionally) and press "Add" button. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.

Modification of an entry for correlation table CC and MCC-MNC

Using CLI

In home routing mode SMSC GW we may need to specify the correlation table CC and MCC-MNC (Correlation table CC and MCC-MNC for home routing mode managing.). You can modify an entry of this table by issuing the command `smsc hrccmccmnc modify` with appropriate parameters as described below.

```
Name
    smsc hrccmccmnc modify

SYNOPSIS
    smsc hrccmccmnc modify <countrycode> <mccmnc> smscgt <smsc-gt>

DESCRIPTION
    This command is used to modify an entry to the correlation
    table CC and MCC-MNC.
    mscgt parameter is optional. If this parameter is missed mscgt
    value will not updated. To set "null" value for <smsc-gt> we
    need to specify "-1" value in CLI.

EXAMPLES
    smsc hrccmccmnc modify 2223 55322 smscgt 733211232342
```

Using GUI

Procedure: Modification of an entry for correlation table CC and MCC-MNC using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Home Routing CC - MCC MNC Table' in the left panel.
2. You can modify an entry of the table - you can specify "Country Code", "Mobile Country Code and Mobile Network Code", "Global Title" (optionally) and press "Update" button. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.

Removing of an entry for correlation table CC and MCC-MNC

Using CLI

In home routing mode SMSC GW we may need to specify the correlation table CC and MCC-MNC (Correlation table CC and MCC-MNC for home routing mode managing.). You can remove an entry of this table by issuing the command `smsc hrccmccmnc remove` with appropriate parameters as described below.

```
Name
    smsc hrccmccmnc remove

SYNOPSIS
    smsc hrccmccmnc remove <countrycode>

DESCRIPTION
    This command is used to remove an entry to the correlation
    table CC and MCC-MNC.

EXAMPLES
    smsc hrccmccmnc remove 2223
```

Using GUI

Procedure: Removing of an entry for correlation table CC and MCC-MNC using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Home Routing CC - MCC MNC Table' in the left panel.
2. You can delete an entry of the table you - can specify "Country Code" and press "Delete" button. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.

Displaying of an entry / full list of correlation table CC and MCC-MNC

Using CLI

In home routing mode SMSC GW we may need to specify the correlation table CC and MCC-MNC (Correlation table CC and MCC-MNC for home routing mode managing.). You can observe an entry or a full list of this table by issuing the command `smsc hrccmccmnc show` with appropriate parameters as described below.

```
Name
    smsc hrccmccmnc show

SYNOPSIS
    smsc hrccmccmnc show <countrycode>

DESCRIPTION
    This command is used to display of content of an entry
    or a full list of the correlation table CC and MCC-MNC.
    <countrycode> is an optional parameter. If you specify
    it data for only this entry will be displayed. If not
    - data for all entries of that table.

EXAMPLES
    smsc hrccmccmnc show 2223
```

Using GUI

Procedure: Displaying of an entry for correlation table CC and MCC-MNC using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Home Routing CC - MCC MNC Table' in the left panel.
2. You can view content of an entry of the table you or of all table. For viewing of one entry you need to specify "Country Code". Then press "View" button. Results will be shown in the bottom of the screen For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.

6.1.7. CDR

CDR generation

Using CLI

SMSC GW can store CDR records for each delivered or failed for delivering messages into log file (see [_cdr_logging_settings]). Which records will be stored is defined by 'generatecdr' SMSC option. You can set the 'generatecdr' option by issuing the command `smsc set generatecdr` with appropriate parameters as described below. You can verify this by issuing the command `smsc get generatecdr` which will display the value set for this property.

```
Name
    smsc set generatecdr

SYNOPSIS
    smsc set generatecdr <digital option>

DESCRIPTION
    This command is used to set which messages (or none) will be stored into
    CDR log file. Details of CDR log format can be found in "8.2. CDR Log"
    Options will have following bits values:
    bit 1 - records will be done for SMPP originated messages with datagramm
            mode
    bit 2 - records will be done for SMPP originated messages with
            transactional mode
    bit 4 - records will be done for SMPP originated messages with
            storeAndForward mode and for all SS7 or SIP originated messages
    Value 0 will mean store none and value 7 - store all.
    Default value: 7 (store all)

EXAMPLES
    smsc set generatecdr 7
```

Using GUI

Procedure: Set CDRs generation SMSC option using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Properties' tab in the GUI.
3. You can specify CDR generation option by selecting an appropriate value in the field "CDR generation". For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

Archive table generation

Using CLI

SMSC GW can store CDR records for each delivered or failed for delivering messages into special archive tables of cassandra database. Names of such tables have the following format: MESSAGES_yyyy_mm_dd. You can refer to fields definitions in [_messages_yyyy_mm_dd] chapter. Which records will be stored is defined by 'generatearchivetable' SMSC option. You can set the 'generatearchivetable' option by issuing the command `smsc set generatearchivetable` with appropriate parameters as described below. You can verify this by issuing the command `smsc get generatearchivetable` which will display the value set for this property.

```
Name
    smsc set generatearchivetable

SYNOPSIS
    smsc set generatearchivetable <digital option>

DESCRIPTION
    This command is used to set which messages (or none) will be stored into
    archive tables. Options will have following bits values:
    bit 1 - records will be done for SMPP originated messages with datagramm
            mode
    bit 2 - records will be done for SMPP originated messages with
            transactional mode
    bit 4 - records will be done for SMPP originated messages with
            storeAndForward mode and for all SS7 or SIP originated messages
    Value 0 will mean store none and value 7 - store all.
    Default value: 7 (store all)
EXAMPLES
    smsc set generatearchivetable 7
```

Using GUI

Procedure: Set archive table CDRs generation SMSC option using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Properties' tab in the GUI.
3. You can specify archive table CDR generation option by selecting an appropriate value in the field "Archive table generation". For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

Generate CDR for Receipt Messages

Using CLI

You can set the 'generatereceiptcdr' value to true/false by issuing the command `smsc set generatereceiptcdr` with appropriate parameters as described below. You can verify this by issuing the command `smsc get generatereceiptcdr` which will display the value set for this property.

```
Name
    smsc set generatereceiptcdr

SYNOPSIS
    smsc set generatereceiptcdr <true | false>

DESCRIPTION
    The SMSC can be configured to generate CDR for both receipt and regular messages
    or generate CDR only for regular messages. By default the SMSC will generate CDR
    for regular messages only. However if you require the SMSC to generate CDR for
    receipt messages as well, you must set the parameter 'generatereceiptcdr' to true.
```

Using GUI

Procedure: Set 'generate CDR for receipt messages' using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Properties' tab in the GUI.
3. You can specify if the SMSC should generate CDR for receipt messages by choosing the value true for the field 'Generate receipt CDR'. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

6.1.8. Processing

SMSC pausing

Using CLI

A user can pause SMSC for message delivering by issuing the command `smsc set deliverypause` with appropriate parameters as described below. When SMSC delivery is paused no more messages are scheduled for delivering. When SMSC is configured in ForwardAndStore (fast) mode all incoming messages will be rejected. When SMSC is configured in StoreAndForward (normal) mode datagramm and transactional SMPP originated messages will be rejected but StoreAndForward SMPP originated and all SS7 / SIP originated messages will be stored into cassandra database without delivery attempts. Use this option with extreme caution.

```
    Name
        smsc set deliverypause

    SYNOPSIS
        smsc set deliverypause <true|false>

    DESCRIPTION
        Setting to true puts SMSC GW into a pause mode and setting to false
        (default value) returns SMSC GW to a normal message processing.
    EXAMPLES
        smsc set deliverypause true
```

Using GUI

Procedure: Pause SMSC using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. You can select "true" or "false" value. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
3. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

Disabling of Reciepts generating

Using CLI

SMCS can generate receipts of messages delivering results (success / error). See chapter "Appendix B. Delivery Receipt Format" of "Short Message Peer to Peer. Protocol Specification v3.4." You can set the 'receiptsdisabling' value to true/false by issuing the command `smsc set receiptsdisabling` with appropriate parameters as described below. You can verify this by issuing the command `smsc get receiptsdisabling` which will display the value set for this property.

```
Name
    smsc set receiptsdisabling

SYNOPSIS
    smsc set receiptsdisabling <true | false>

DESCRIPTION
    The SMSC can be configured to generat or not delivery receipts.
     Setting of receiptsdisabling to false enables of receipts generation.
     Setting of receiptsdisabling to true disables of receipts generation.
     Default value: false (receipts will be generated).
```

Using GUI

Procedure: Disabling of Reciepts generating using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Processing' tab in the GUI.
3. You can specify if the SMSC should disable of receipt generating by setting the value for the field 'Disable Delivery Receipt' to true or false accordingly. For more details of this parameter, please refer to the description of the CLI command for the same in the preceding section.
4. You must click on the button 'Save' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

Routing of delivery receipts

Using CLI

If this option is turned on for each generated delivery receipt networkId will be assigned to the value of networkId of an ESME via which the original message has come to SMSC. This can help for routing of receipts back to the originated ESME. If this option is turned off then the networkId of receipts will be taken from networkId of ESME / SS7 / SIP via which the original message has left SMSC GW. You can configure this option by issuing the command smsc set orignetworkidforreceipts with appropriate parameters as described below.

```
Name
    smsc set orignetworkidforreceipts

SYNOPSIS
    smsc set orignetworkidforreceipts <true | false>

DESCRIPTION

    Settings of this option will affect of which networkId will be assigned
    to a message delivery receipt.
    true: networkId of the connector via which an original message has
          left SMSC GW
    false: networkId of the connector via which an original message has
           come SMSC GW. This value is default.
```

Using GUI

Procedure: Routing of delivery receipts option update by the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Server Settings' in the left panel.
2. The main panel will display the existing Settings, segregated into eight horizontal tabs: Properties, SS7 Settings, Cassandra, Scheduler, Diameter, Processing, CDR and Home Routing. Switch to the 'Processing' tab in the GUI.
3. You can specify routing of delivery receipts by setting the value for the field 'Delivery receipts will be routed to the origination networkId' to true or false accordingly. For more details of this

parameter, please refer to the description of the CLI command for the same in the preceding section.

4. You must click on the button 'Save' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

6.2. SMPP Server Settings

6.2.1. View SMPP Server Details

Using GUI

Procedure: View SMPP Server Details using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'SMPP Server' in the left panel.
2. The main panel will display the existing SMPP Server details as configured in the section `SmscManagement` in the xml descriptor file *restcomm-smsc-<version>/jboss-5.1.0.GA/server/<profile>/deploy/restcomm-smsc-server/META-INF/jboss-beans.xml*.
3. You can view the current Session details of the SMPP Server by clicking on the row corresponding to the name of the SMPP Server (`SmscManagement` in this case).

6.2.2. Edit SMPP Server Properties

Using CLI

You can edit the properties of the SMPP Server by issuing appropriate commands for every property as described below:

```
Name
    smppserver set port

SYNOPSIS
    smppserver set port <port>

DESCRIPTION
    This command is used to set the port that the SMSC server is listening to for
    incoming bind request.

    If unspecified, the default port is 2776.

    You must restart the SMPP Server for the new value to take effect.
```

```
Name
    smppserver set bindtimeout

SYNOPSIS
    smppserver set bindtimeout <bind-timeout>

DESCRIPTION
    This command is used to set a value for bind-timeout in milli-seconds.
    Once the TCP socket is established, the SMSC server will wait for the time period
    specified by the parameter bind-timeout in milli-seconds, for the peer to send a
    bind request, after which it will kill the TCP socket.

    If unspecified, the default value is 5000 milli-seconds.

    You must restart the SMPP Server for the new value to take effect.
```

```
Name
    smppserver set systemid

SYNOPSIS
    smppserver set systemid <system-id>

DESCRIPTION
    This command is used to set the value for system-id. This is the 'system-id'
    included in the Bind response.

    You must restart the SMPP Server for the new value to take effect.
```

```
Name
    smppserver set autonegotiateversion

SYNOPSIS
    smppserver set autonegotiateversion <true/false>

DESCRIPTION
    This command is used to specify if auto-negotiate-version is enabled or not.
    If it is set to 'true' and a Bind is received with version <= 3.3 for
    interface version, then it is normalized to version 3.3. If a Bind is received
    with version >= 3.4 for interface version, it is normalized to version 3.4.

    The default value is true.

    You must restart the SMPP Server for the new value to take effect.
```

```
Name
    smppserver set interfaceversion

SYNOPSIS
    smppserver set interfaceversion <interface-version>

DESCRIPTION
    This command is used to specify the SMPP version that the Server supports.

    You must restart the SMPP Server for the new value to take effect.
```

```
Name
    smppserver set maxconnectionsize

SYNOPSIS
    smppserver set maxconnectionsize <max-connection-size>

DESCRIPTION
    This command is used to specify the maximum number of connections/sessions this
    Server is expected to handle.

    You must restart the SMPP Server for the new value to take effect.
```

```
Name
    smppserver set defaultwindowsize

SYNOPSIS
    smppserver set defaultwindowsize <defaultwindowsize>

DESCRIPTION
    This command is used to specify the default window size for this Server.
    The window size is the amount of unacknowledged requests that are permitted to be
    outstanding/unacknowledged at any given time. If more requests are added, the
    underlying stack will throw an exception.

    The default value is 100.

    You must restart the SMPP Server for the new value to take effect.
```

```
Name
    smppserver set defaultwindowwaittimeout

SYNOPSIS
    smppserver set defaultwindowwaittimeout <default-window-wait-timeout>

DESCRIPTION
    This command is used to specify the default-window-wait-timeout for this Server
    in milli-seconds.
    The window wait timeout is the time within which the connection to remote SMSC
    Server should be established.

    The default value is 30000 milli seconds.

    You must restart the SMPP Server for the new value to take effect.
```

```
Name
    smppserver set defaultrequestexpirytimeout

SYNOPSIS
    smppserver set defaultrequestexpirytimeout <default-request-expiry-timeout>

DESCRIPTION
    This command is used to specify the default-request-expiry-timeout for the Server
    in milli-seconds. The request expiry timeout is the time to wait for an end-point
    to respond to before it expires.

    The default value is 30000 milli seconds.

    You must restart the SMPP Server for the new value to take effect.
```

```
Name
    smppserver set defaultwindowmonitorinterval

SYNOPSIS
    smppserver set defaultwindowmonitorinterval <default-window-monitor-interval>

DESCRIPTION
    This command is used to specify the default-window-monitor-interval for the Server
    in milli-seconds. This is the time between executions of monitoring the window for
    requests that expire. It is recommended that this value, generally, either matches
    or is half the value of 'request-expiry-timeout'. Therefore, in the worst case
    scenario, a request could take upto 1.5 times the 'requestExpiryTimeout' to
    clear out.

    The default value is 15000 milli seconds.

    You must restart the SMPP Server for the new value to take effect.
```

```
Name
    smppserver set defaultsessioncountersenabled

SYNOPSIS
    smppserver set defaultsessioncountersenabled <true/false>

DESCRIPTION
    This command is used to set the parameter 'defaultsessioncountersenabled' value
    to true or false.
    When this is enabled, SMSC exposes the statistics for SMPP connections.

    The default value is true.

    You must restart the SMPP Server for the new value to take effect.
```

Using GUI

Procedure: Edit SMPP Server Properties using GUI

1. In the GUI Management Console for SMSC Gateway, click on 'SMPP Server' in the left panel.
2. The main panel will display the existing SMPP Server details as configured in the section `SmscManagement` in the xml descriptor file *restcomm-smsc-<version>/jboss-5.1.0.GA/server/<profile>/deploy/restcomm-smsc-server/META-INF/jboss-beans.xml*.
3. You can edit the properties of the SMPP Server by launching the edit window. You can achieve this by clicking on the blue coloured 'edit' button at the end of the row. The edit window will display all SMPP properies as shown in the figure below. For more details of these parameters please refer to the descriptions of the CLI commands for the same in the preceding section.

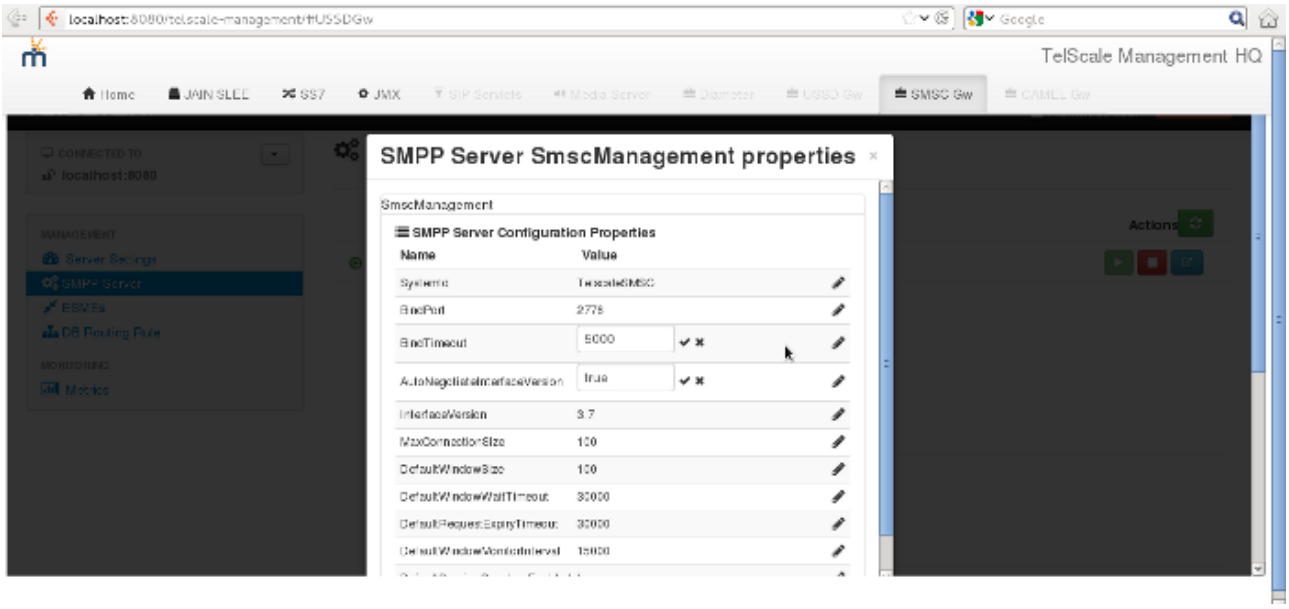


Figure 2. SMPP Server - GUI - Restcomm SMSC

4. To edit any property, click on the edit icon of the row corresponding to the property. This action will display an editable text field for the property as shown in the figure above. Adjacent to the editable text field, you will find a 'tick' icon and a 'x' icon. To accept the newly entered value for the property, you must click on the 'tick' icon. To discard the change and stop the editing of the property, you must click on the 'x' icon.

   SMPP Server can be setup for SSL so every incoming connection request should first do SSL hand-shake. Settingup SSL is only possible from GUI.

5. You must click on the button 'Apply Changes' at the top of the window to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.

6.2.3. Start SMPP Server

Using GUI

Procedure: Start SMPP Server using GUI

1. In the GUI Management Console for SMSC Gateway, click on 'SMPP Server' in the left panel.
2. The main panel will display the existing SMPP Server details as configured in the section `SmscManagement` in the xml descriptor file *restcomm-smsc-<version>/jboss-5.1.0.GA/server/<profile>/deploy/restcomm-smsc-server/META-INF/jboss-beans.xml*.
3. You can start the SMPP Server by clicking on the 'Start' icon lit green in the row corresponding to the SmscManagement unit. This icon will be enabled only if the SMPP server is currently stopped.
4. This action will start the SMPP Server.
5. If there is an error in starting the SMPP Server, then you will find the details of the error in the Management Console Log section below.

6.2.4. Stop SMPP Server

Using GUI

Procedure: Stop SMPP Server using GUI

1. In the GUI Management Console for SMSC Gateway, click on 'SMPP Server' in the left panel.
2. The main panel will display the existing SMPP Server details as configured in the section `SmscManagement` in the xml descriptor file *restcomm-smsc-<version>/jboss-5.1.0.GA/server/<profile>/deploy/restcomm-smsc-server/META-INF/jboss-beans.xml*.
3. You can stop the SMPP Server by clicking on the 'Stop' icon lit red in the row corresponding to the SmscManagement unit. This icon will be enabled only if the SMPP server is currently running.
4. This action will stop the SMPP Server.
5. If there is an error in stopping the SMPP Server, then you will find the details of the error in the Management Console Log section below.

6.2.5. Reset Counters for SMPP Server

Using GUI

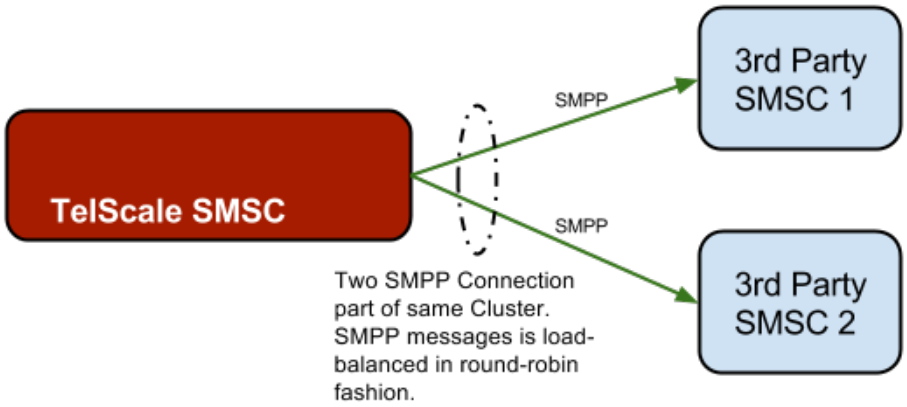
Procedure: Reset Counters for SMPP Server using GUI

1. In the GUI Management Console for SMSC Gateway, click on 'SMPP Server' in the left panel.
2. The main panel will display the existing SMPP Server details as configured in the section `SmscManagement` in the xml descriptor file *restcomm-smsc-<version>/jboss-5.1.0.GA/server/<profile>/deploy/restcomm-smsc-server/META-INF/jboss-beans.xml*.
3. You can view the current Session details of the SMPP Server by clicking on the row corresponding to the name of the SMPP Server (`SmscManagement` in this case).
4. This action will display the current session details of the SMPP Server. If you scroll to the bottom, you will find a button named 'Reset Counters'. Click on it if you wish to reset all counters for SMPP Server.
5. If there is an error resetting the counters, then you will find the details of the error in the Management Console Log section below.

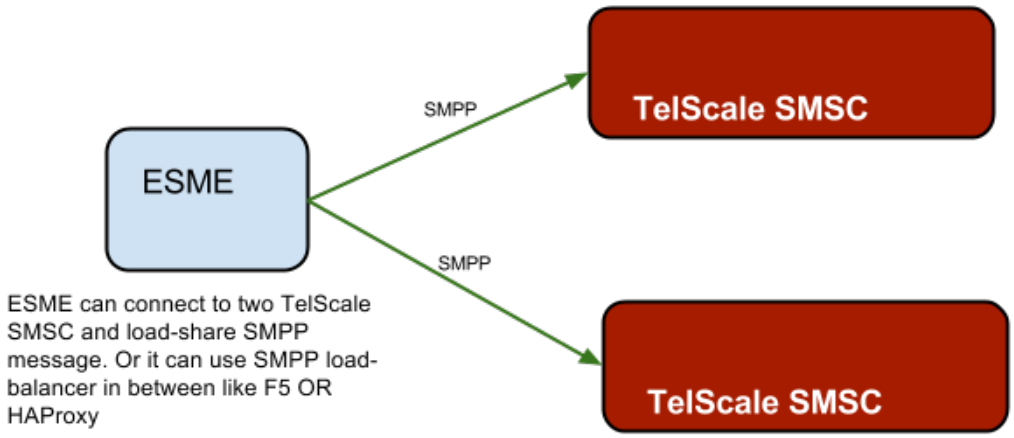
6.3. External Short Messaging Entities (ESMEs)

&THIS.PLATFORM;&THIS.APPLICATION;can now act as ESME (initiate bind to remote SMSC) or can also act as SMSC (accept bind from remote ESME). While defining an ESME (SMPP connection), you can optionally pass the Cluster name. If it is not passed, cluster name is same as ESME name. As the name suggests, its now possible to group different ESMEs in the same cluster. This is useful only when SMS is suppose to be routed out of &THIS.PLATFORM;&THIS.APPLICATION;to ESME. If there are multiple ESME's in a cluster, the load is shared in a round robin fashion to send out SMS. In case if one of the ESME is in `UNBOUND` state, the next `BOUND` smpp connection (within same cluster) will be used. Below diagram explains the load-balancing between SMPP connections

Case 1 : SMSC acting as CLIENT



Case 2 : SMSC acting as SERVER



Case 3 : SMSC acting as SERVER - Load balancing via HAProxy

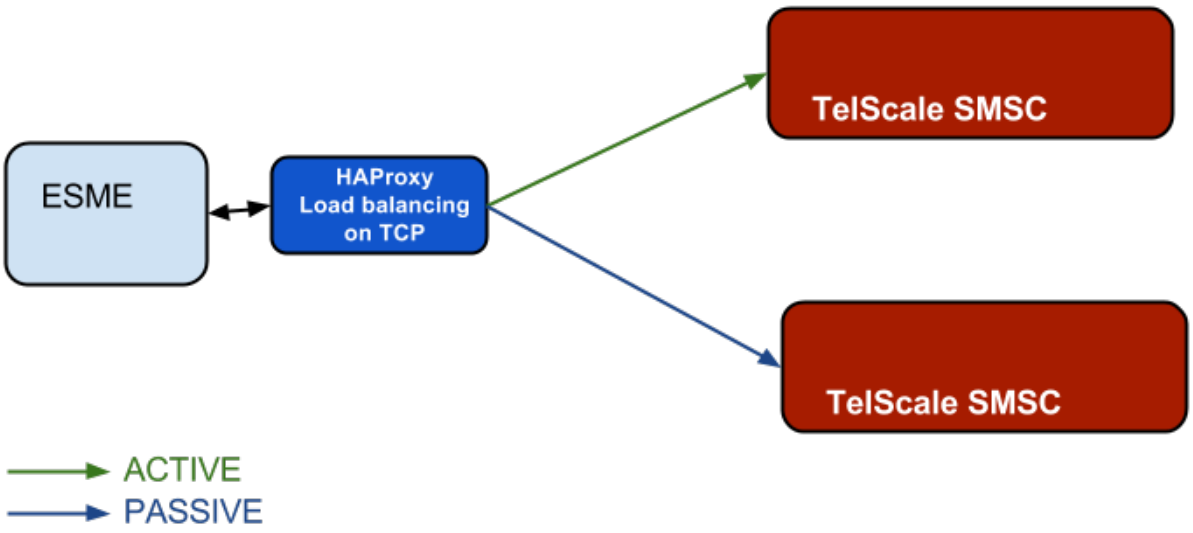


Figure 3. SMPP Load Balancing

You can define multiple ESMEs, each with a unique name but the same `systemId` and declare their `host-ip` and `port` values as -1 (only for SMPP type Server). The SMSC will now accept multiple incoming BIND requests from any `IP` and any `port` as far as the `systemId` and `Password` matches.

Alternatively, if you declare a real value for `host-ip`(say for example 10.199.7.23) and `port` as -1, the SMSC will accept as many BINDS as ESMEs defined from the specified IP but any port as far as the `systemId` and `Password` matches.

6.3.1. Create new ESME

Using CLI

You can configure a new ESME by issuing the command ` smpp esme create` with appropriate parameters as described below.

```
Name
    smpp esme create

SYNOPSIS
    smpp esme create name <systemId> <host-ip> <port> <SmppBindType>
    <SmppSession.Type> password <password> networkid <networkId>
    system-type <sms | vms | ota >
    interface-version <3.3 | 3.4 | 5.0> esme-ton <esme address ton>
    esme-npi <esme address npi> esme-range <esme address range>
    cluster-name <clusterName> window-size <windowSize>
    connect-timeout <connectTimeout> request-expiry-timeout <requestExpiryTimeout>
    window-monitor-interval <windowMonitorInterval>
    window-wait-timeout <windowWaitTimeout> counters-enabled <true | false>
    enquire-link-delay <30000> enquire-link-delay-server <0> charging-enabled <true |
false>
    source-ton <source address ton> source-npi <source address npi>
    source-range <source address range> routing-ton <routing address ton>
    routing-npi <routing address npi> routing-range <routing address range>
    ratelimit-second <ratelimitsecond> ratelimit-minute <ratelimitminute>
    ratelimit-hour <ratelimithour> ratelimit-day <ratelimitday>
    min-message-length <min-message-length value>
    max-message-length <max-message-length value>
    national-language-locking-shift <NationalLanguageIdentifier value>
    national-language-single-shift <NationalLanguageIdentifier value>

DESCRIPTION
    This command is used to configure a new ESME.

PARAMETERS

    Standard Parameters

    Name            - A unique name for this ESME configuration. You can
```

```
                define as many ESMEs as you want as far as the name is
                unique and the combination of
                SystemId:host-ip:port:SmppBindType is unique.

    System Id       - This is used to identify an ESME or an SMSC at bind
                time. An 'ESME system_id' identifies the ESME or ESME
                agent to the SMSC. The 'SMSC system_id' provides an
                identification of the SMSC to the ESME. You can define
                multiple ESMEs, each with a unique name but the same
                <literal>systemId<literal> to allow anonymous incoming
                binds and multiple binds from the same IP depending on
                the values specified for host-ip and port.

    host-ip & port      - If the SMSC is acting as an ESME, the BIND request will
                be sent to the configured IP and Port. If the SMSC is
                acting as a Server, it will accept incoming BIND
                requests from the specified IP and Port.
                If the port is unknown, you must pass '-1' as wild
                character.

                When you define multiple ESMEs with the same systemId,if
                host-ip and port values are -1 (for SMPP type Server),
                the SMSC will accept multiple incoming BIND requests
                    from any IP:port as long as the systemId and password
                match.

                When you define multiple ESMEs with the same systemId,if
                host-ip is a real value (a specific IP) and port value
                is -1, the SMSC will accept as many BINDS as ESMEs
                defined from the specified IP but any port as long as
                the systemId and password match.

    SmppBindType        - Possible values: TRANSCEIVER, TRANSMITTER or RECEIVER.
                If the SMSC is acting as an ESME, it will initiate
                corresponding bind. If the SMSC is acting as a Server,
                it will accept corresponding bind from a remote ESME.

    SmppSession.Type    - Possible values: SERVER or CLIENT.
                If the value is 'SERVER', the SMSC acts as a Server
                listening for incoming SMPP binds.
                If the value is 'CLIENT', the SMSC will initiate
                SMPP bind to a remote Server.

    Optional Parameters

    Password        - It is used by the SMSC to authenticate the identity of
                the binding ESME. The Service Provider may require
                ESME's to provide a password when binding to the SMSC.

    networkId       - indicates virtual subnetwork that this ESME belongs. SMS
                flows within same netwrokId, unless changed using mproc
```

```
                (this is for multi-tenancy support). If this parameter is
                skipped - networkId will be set to "0" when ESME creation.
                If you do not use multi-tenancy support - set this value
                to 0 or skip.

system-type         - Default value is null.
              This is used to categorize the type of ESME that is
              binding to the SMSC.

interface-version   - Default value is 3.4.

              It is used to indicate the version of the SMPP protocol.

              It is set in 'SMPPServer Settings'.

esme-ton        - Defines ESME TON. If the SMPP Session Type is CLIENT,
              this TON will be used in the BIND request. If the SMPP
              Session Type is SERVER, the incoming BIND request should
              have the same TON as configured here. If the configured
              value is null (-1), SMSC will ignore it in both cases.

esme-npi        - Defines ESME NPI. If the SMPP Session Type is CLIENT,
              this NPI will be used in the BIND request. If the SMPP
              Session Type is SERVER, the incoming BIND request should
              have the same NPI as configured here. If the configured
              value is null (-1), SMSC will ignore it in both cases.

esme-range      - Defines ESME Address Range. If the SMPP Session Type is
              CLIENT, this Address Range will be used in the BIND
              request. If the SMPP Session Type is SERVER, the
              incoming BIND request should have the same Address Range
              as configured here. If the configured value is
              null (-1), SMSC will ignore it in both cases.

cluster-name        - If it is not specified then its same as the name.
              It is possible to group different SMPP connections
              together by specifying the same cluster-name.
              All the SMPP connection's that are capable of sending
              out SMS are candidates for grouping.

window-size         - Default value is 1.
              The window size is the amount of unacknowledged requests
              that are permitted to be outstanding/unacknowledged at
              any given time. If more requests are added, the
              underlying stack will throw an exception.

              This value is set only when ESME is defined as Client
              side. For Server side this value is taken from the
              'SMPP Server Settings'.

connect-timeout     - Default value is 10000 milli seconds.
```

```
              This parameter is used to specify the time within which
              the connection to a remote SMSC server should be
              established.

              This is useful only when ESME is defined as Client Side.
              For Server side this value is taken from the
              the 'SMPP Server Settings'.

request-expiry-timeout  - Default value is -1 (disabled).
              This parameter is used to specify the time to wait in
              milli seconds for an endpoint to respond to before it
              expires.

              This is useful only when ESME is defined as Client Side.
              For Server side this value is taken from the
              the 'SMPP Server Settings'.

window-monitor-interval - Default value is -1 (disabled).
              This parameter is used to specify the time between
              executions of monitoring the window for requests that
              expire. It is recommended that this value, generally,
              either matches or is half the value of
              'request-expiry-timeout'. Therefore, in the worst case
              scenario, a request could take upto 1.5 times the
              'requestExpiryTimeout' to clear out.

              This is useful only when ESME is defined as Client Side.
              For Server side this value is taken from the
              the 'SMPP Server Settings'.

window-wait-timeout     - Default value is 60000 milli seconds.
              This parameter is used to specify the time to wait
              until a slot opens up in the 'sendWindow'.

              This is useful only when ESME is defined as Client Side.
              For Server side this value is taken from the
              the 'SMPP Server Settings'.

counters-enabled    - Default value is true.
              When this is enabled, SMSC exposes the statistics for
              SMPP connections.

              This is useful only when ESME is defined as Client Side.
              For Server side this value is taken from the
              the 'SMPP Server Settings'.

enquire-link-delay  - Default value is 30000 milli seconds.
              When SMSC connects to a remote server as CLIENT, it
              sends an 'ENQUIRE_LINK' after every configured
              enquire-link-delay.
```

```
enquire-link-delay-server - Default value is 0 milli seconds.
              When SMSC connects to a remote server as SERVER, it
              sends an 'ENQUIRE_LINK' after every configured
              enquire-link-delay-server.

charging-enabled    - Flag to enable or disable charging for every SMS
              arriving from SIP.

source-ton      - Every SMS coming into the SMSC via this ESME should have
              the same 'source_addr_ton' as the value configured here.

              If this configured value is null(-1) or not null and
              matches, the SMSC will compare the
              'source_addr_npi' and 'source_addr_range' as explained
              below.

              If it doesn't match, the SMSC will reject this SMS with
              an error code '0x0000000A' indicating
              Invalid Source Address.

source-npi      - Every SMS coming into the SMSC via this ESME should have
              the same 'source_addr_npi' as the value configured here.
              configured here.

              If this configured value is null(-1)
              or not null and matches, the SMSC will compare the
              'source_addr_range' as below.

              If it doesn't match, the SMSC will reject this
              SMS with an error code '0x0000000A' indicating
              Invalid Source Address.

source-range        - Every SMS coming into the SMSC via this ESME should have
              the same 'source_addr_range' as the value configured
              here.This is a regular java expression and
              default value is ^[0-9a-zA-Z]*.

              If it matches, the SMSC will accept the incoming SMS
              and process further.

              If it doesn't match, the SMSC will reject this
              SMS with an error code '0x0000000A' indicating
              Invalid Source Address.

routing-ton     - The DefaultSmsRoutingRule will try to match the
              'dest_addr_ton' of outgoing SMS with the value
              configured here. If this configured value is null(-1)
              or not null and matches, the SMSC will compare the
              'dest_addr_npi' and 'destination_addr' as explained
              below. If it doesn't match, the SMSC will select the
              next ESME in the list for matching routing rule.
```

```
                DefaultSmsRoutingRule will consider ESME for routing
                only if
                1) SmppBindType is TRANSCEIVER
                2) SmppBindType is RECEIVER and
                SmppSession.Type is SERVER
                3) SmppBindType is TRANSMITTER and
                SmppSession.Type is CLIENT

routing-npi     - The DefaultSmsRoutingRule will try to match the
                'dest_addr_npi' of outgoing SMS with the value
                configured here. If this configured value is null(-1)
                or not null and matches, the SMSC will compare the
                'destination_addr' as below. If it doesn't match, the
                SMSC will select the next ESME in the list for matching
                routing rule.

                DefaultSmsRoutingRule will consider ESME for routing
                only if
                1) SmppBindType is TRANSCEIVER
                2) SmppBindType is RECEIVER and
                SmppSession.Type is SERVER
                3) SmppBindType is TRANSMITTER and
                SmppSession.Type is CLIENT

routing-range       - The DefaultSmsRoutingRule will try to match the
                'destination_addr' of outgoing SMS with the value
                configured here. This is a regular java expression and
                default value is ^[0-9a-zA-Z]*. If it matches, the SMSC
                will send the SMS out over this SMPP connection. If it
                doesn't match, the SMSC will select the next ESME in
                the list for matching routing rule.

                DefaultSmsRoutingRule will consider ESME for routing
                only if
                1) SmppBindType is TRANSCEIVER
                2) SmppBindType is RECEIVER and
                SmppSession.Type is SERVER
                3) SmppBindType is TRANSMITTER and
                SmppSession.Type is CLIENT

ratelimit_second    - This parameter is used to specify a maximum limit
                  of messages that the SMSC will accept from this ESME
                  during any one second.

                  If the ESME sends more messages (per second)
                  than the maximum limit specified by 'ratelimit_second',
                  these additional messages will be rejected by the
                  SMSC GW along with an error code - "throttled".

                  The default value for this parameter is "0" and it
```

```
                implies "no restrictions". If this parameter is not
                specified it implies "no restrictions".

    ratelimit_minute    - This parameter is used to specify a maximum limit
                of messages that the SMSC will accept from this ESME
                during any one minute.

                If the ESME sends more messages (per minute)
                than the maximum limit specified by 'ratelimit_minute',
                these additional messages will be rejected by the
                SMSC GW along with an error code - "throttled".

                The default value for this parameter is "0" and it
                implies "no restrictions". If this parameter is not
                specified it implies "no restrictions".

    ratelimit_hour    - This parameter is used to specify a maximum limit
                of messages that the SMSC will accept from this ESME
                during any one hour.

                If the ESME sends more messages (per hour)
                than the maximum limit specified by 'ratelimit_hour',
                these additional messages will be rejected by the
                SMSC GW along with an error code - "throttled".

                The default value for this parameter is "0" and it
                implies "no restrictions". If this parameter is not
                specified it implies "no restrictions".

    ratelimit_day    - This parameter is used to specify a maximum limit
                of messages that the SMSC will accept from this ESME
                during any one day.

                If the ESME sends more messages (per day)
                than the maximum limit specified by 'ratelimit_day',
                these additional messages will be rejected by the
                SMSC GW along with an error code - "throttled".

                The default value for this parameter is "0" and it
                implies "no restrictions". If this parameter is not
                specified it implies "no restrictions".

    min-message-length    - This paramter is used to specify the minimum
                message length (in characters) acceptable to
                the SMSC GW, for messages coming from this ESME.

                If an incoming message length is less than the
                min-message-length it will be rejected by SMSC GW.

                The default value for this parameter is "-1" and it
                implies "no limitations". Any other negative value
```

```
                  also implies "no limitations".

    max-message-length    - This paramter is used to specify the maximum
                  message length (in characters) acceptable to
                  the SMSC GW, for messages coming from this ESME.

                  If an incoming message length is more than the
                  max-message-length it will be rejected by SMSC GW.

                  The default value for this parameter is "-1" and it
                  implies "no limitations". Any other negative value
                  also implies "no limitations".

    national-language-locking-shift - National language locking shift
                  table can be configured for messages that have come
                  via SMPP (this ESME), do not have UDHs inside and
                  have GSM7 encoding (DCS==0).
                  The default GSM data coding table is mostly used.
                  Possible values:
                  = 0: default GSM data coding table
                  = 13: urdu (arabic) national language shift table
                  =1: the national language locking shift value must
                  be obtained from the option
                  national-language-locking-shift that is defined at
                  SMSC GW general level.

    national-language-single-shift - National language single shift
                  table can be configured for messages that have come
                  via SMPP (this ESME), do not have UDHs inside and
                  have GSM7 encoding (DCS==0).
                  The default GSM data coding table is mostly used.
                  Possible values:
                  = 0: default GSM data coding table
                  = 13: urdu (arabic) national language single table
                  =1: the national language locking shift value must
                  be obtained from the option
                  national-language-locking-single that is defined at
                  SMSC GW general level.
```

Using GUI

Procedure: Create new ESME using GUI

1. In the GUI Management Console for SMSC Gateway, click on 'ESMEs' in the left panel.
2. The main panel will display the existing ESMEs (if any), one each in a row with corresponding actions (start, stop, delete, update) for each row. Below this you will find the button 'Create ESME'.
3. You can create a new ESME by launching the 'Create ESME' window by clicking on the blue coloured 'Create ESME' button. The 'Create ESME' window will display all ESME paramters that must be defined by you. For more details of these parameters please refer to the descriptions of

the CLI commands for the same in the preceding section.

4. Enter appropriate values for all the parameters and then click on the 'Create' button at the bottom of this 'Create ESME' window. This action will create a new ESME with parameters as defined by you.
5. If there is an error in defining the ESME, then you will find the details of the error in the Management Console Log section below.

6.3.2. Modify ESME

Using CLI

You can modify an existing ESME by issuing the command `smpp esme modify` with appropriate parameters as described below.

```
Name
    smpp esme modify

SYNOPSIS
    smpp esme modify <name> password <Specify new password>
    networkid <networkId>
    esme-ton <esme address ton> esme-npi <esme address npi>
    esme-range <esme address range> window-size <windowSize>
    connect-timeout <connectTimeout> request-expiry-timeout <requestExpiryTimeout>
    window-monitor-interval <windowMonitorInterval>
    window-wait-timeout <windowWaitTimeout> counters-enabled <true | false>
    enquire-link-delay <30000> enquire-link-delay-server <0> charging-enabled <true |
false>
    source-ton <source address ton> source-npi <source address npi>
    source-range <source address range> routing-ton <routing address ton>
    routing-npi <routing address npi> routing-range <routing address range>
    ratelimit-second <ratelimitsecond> ratelimit-minute <ratelimitminute>
    ratelimit-hour <ratelimithour> ratelimit-day <ratelimitday>
    min-message-length <min-message-length value>
    max-message-length <max-message-length value>
    national-language-locking-shift <NationalLanguageIdentifier value>
    national-language-single-shift <NationalLanguageIdentifier value>

DESCRIPTION
    This command is used to modify the settings of an existing ESME configuration.

PARAMETERS

    Standard Parameters

    Name            - The name of the ESME that is being modified.

    Optional Parameters

    Password        - Specify the new password.
```

```
                It is used by the SMSC to authenticate the identity of
                the binding ESME. The Service Provider may require
                ESMEs to provide a password when binding to the SMSC.

                The new value takes effect when SMPP is restarted.

networkId         - indicates virtual subnetwork that this ESME belongs. SMS
                  flows within same netwrokId, unless changed using mproc
                  (this is for multi-tenancy support). If this parameter is
                  skipped - networkId will be set to "0" when ESME creation.
                  If you do not use multi-tenancy support - set this value
                  to 0 or skip.

esme-ton          - Specify new ESME TON.
                If the SMPP Session Type is CLIENT,
                this TON will be used in the BIND request. If the SMPP
                Session Type is SERVER, the incoming BIND request should
                have the same TON as configured here. If the configured
                value is null (-1), SMSC will ignore it in both cases.

                The new value takes effect when SMPP is restarted.

esme-npi          - Specify new ESME NPI.
                If the SMPP Session Type is CLIENT,
                this NPI will be used in the BIND request. If the SMPP
                Session Type is SERVER, the incoming BIND request should
                have the same NPI as configured here. If the configured
                value is null (-1), SMSC will ignore it in both cases.

                The new value takes effect when SMPP is restarted.

esme-range        - Specify ESME Address Range.
                If the SMPP Session Type is
                CLIENT, this Address Range will be used in the BIND
                request. If the SMPP Session Type is SERVER, the
                incoming BIND request should have the same Address Range
                as configured here. If the configured value is
                null (-1), SMSC will ignore it in both cases.

                The new value takes effect when SMPP is restarted.

window-size           - Specify new window size.
                Default value is 1.
                The window size is the amount of unacknowledged requests
                that are permitted to be outstanding/unacknowledged at
                any given time. If more requests are added, the
                underlying stack will throw an exception.

                This value is set only when ESME is defined as Client
                side. For Server side this value is taken from the
                'SMPP Server Settings'.
```

```
                The new value takes effect when SMPP is restarted.

    connect-timeout     - Default value is 10000 milli seconds.
                This parameter is used to specify the time within which
                the connection to a remote SMSC server should be
                established.

                This is useful only when ESME is defined as Client Side.
                For Server side this value is taken from the
                the 'SMPP Server Settings'.

                The new value takes effect when SMPP is restarted.

    request-expiry-timeout  - Default value is -1 (disabled).
                This parameter is used to specify the time to wait in
                milli seconds for an endpoint to respond to before it
                expires.

                This is useful only when ESME is defined as Client Side.
                For Server side this value is taken from the
                the 'SMPP Server Settings'.

                The new value takes effect when SMPP is restarted.

    window-monitor-interval - Default value is -1 (disabled).
                This parameter is used to specify the time between
                executions of monitoring the window for requests that
                expire. It is recommended that this value, generally,
                either matches or is half the value of
                'request-expiry-timeout'. Therefore, in the worst case
                scenario, a request could take upto 1.5 times the
                'requestExpiryTimeout' to clear out.

                This is useful only when ESME is defined as Client Side.
                For Server side this value is taken from the
                the 'SMPP Server Settings'.

                The new value takes effect when SMPP is restarted.

    window-wait-timeout     - Default value is 60000 milli seconds.
                This parameter is used to specify the time to wait
                until a slot opens up in the 'sendWindow'.

                This is useful only when ESME is defined as Client Side.
                For Server side this value is taken from the
                the 'SMPP Server Settings'.

                The new value takes effect when SMPP is restarted.

    counters-enabled    - Default value is true.
```

```
                When this is enabled, SMSC exposes the statistics for
                SMPP connections.

                This is useful only when ESME is defined as Client Side.
                For Server side this value is taken from the
                the 'SMPP Server Settings'.

                The new value takes effect when SMPP is restarted.

    enquire-link-delay  - Default value is 30000 milli seconds.
                When SMSC connects to a remote server as CLIENT, it
                sends an 'ENQUIRE_LINK' after every configured
                enquire-link-delay.

                0 means disabled. SMSC will not send ENQUIRE_LINK.

                The new value takes effect immediately.

    enquire-link-delay-server - Default value is 0 milli seconds.
                When SMSC connects to a remote server as SERVER, it
                sends an 'ENQUIRE_LINK' after every configured
                enquire-link-delay-server.

                0 means disabled. SMSC will not send ENQUIRE_LINK.

                The new value takes effect immediately.

    charging-enabled    - Flag to enable or disable charging for every SMS
                arriving from SIP.

                The new value takes effect immediately.

    source-ton      - Every SMS coming into the SMSC via this ESME should have
                the same 'source_addr_ton' as the value configured here.

                If this configured value is null(-1) or not null and
                matches, the SMSC will compare the
                'source_addr_npi' and 'source_addr_range' as explained
                below.

                If it doesn't match, the SMSC will reject this SMS with
                an error code '0x0000000A' indicating
                Invalid Source Address.

                The new value takes effect immediately.

    source-npi      - Every SMS coming into the SMSC via this ESME should have
                the same 'source_addr_npi' as the value configured here.
                configured here.

                If this configured value is null(-1)
```

```
                or not null and matches, the SMSC will compare the
                'source_addr_range' as below.

                If it doesn't match, the SMSC will reject this
                SMS with an error code '0x0000000A' indicating
                Invalid Source Address.

                The new value takes effect immediately.

    source-range        - Every SMS coming into the SMSC via this ESME should have
                the same 'source_addr_range' as the value configured
                here.This is a regular java expression and
                default value is ^[0-9a-zA-Z]*.

                If it matches, the SMSC will accept the incoming SMS
                and process further.

                If it doesn't match, the SMSC will reject this
                SMS with an error code '0x0000000A' indicating
                Invalid Source Address.

                The new value takes effect immediately.

    routing-ton     - The DefaultSmsRoutingRule will try to match the
                'dest_addr_ton' of outgoing SMS with the value
                configured here. If this configured value is null(-1)
                or not null and matches, the SMSC will compare the
                'dest_addr_npi' and 'destination_addr' as explained
                below. If it doesn't match, the SMSC will select the
                next ESME in the list for matching routing rule.

                DefaultSmsRoutingRule will consider ESME for routing
                only if
                1) SmppBindType is TRANSCEIVER
                2) SmppBindType is RECEIVER and
                SmppSession.Type is SERVER
                3) SmppBindType is TRANSMITTER and
                SmppSession.Type is CLIENT

                The new value takes effect immediately.

    routing-npi     - The DefaultSmsRoutingRule will try to match the
                'dest_addr_npi' of outgoing SMS with the value
                configured here. If this configured value is null(-1)
                or not null and matches, the SMSC will compare the
                'destination_addr' as below. If it doesn't match, the
                SMSC will select the next ESME in the list for matching
                routing rule.

                DefaultSmsRoutingRule will consider ESME for routing
                only if
```

```
                1) SmppBindType is TRANSCEIVER
                2) SmppBindType is RECEIVER and
                SmppSession.Type is SERVER
                3) SmppBindType is TRANSMITTER and
                SmppSession.Type is CLIENT

                The new value takes effect immediately.

routing-range       - The DefaultSmsRoutingRule will try to match the
                'destination_addr' of outgoing SMS with the value
                configured here. This is a regular java expression and
                default value is ^[0-9a-zA-Z]*. If it matches, the SMSC
                will send the SMS out over this SMPP connection. If it
                doesn't match, the SMSC will select the next ESME in
                the list for matching routing rule.

                DefaultSmsRoutingRule will consider ESME for routing
                only if
                1) SmppBindType is TRANSCEIVER
                2) SmppBindType is RECEIVER and
                SmppSession.Type is SERVER
                3) SmppBindType is TRANSMITTER and
                SmppSession.Type is CLIENT

                The new value takes effect immediately.

ratelimit_second    - This parameter is used to specify a maximum limit
                  of messages that the SMSC will accept from this ESME
                  during any one second.

                  If the ESME sends more messages (per second)
                  than the maximum limit specified by 'ratelimit_second',
                  these additional messages will be rejected by the
                  SMSC GW along with an error code - "throttled".

                  The default value for this parameter is "0" and it
                  implies "no restrictions". If this parameter is not
                  specified it implies "no restrictions".

ratelimit_minute    - This parameter is used to specify a maximum limit
                  of messages that the SMSC will accept from this ESME
                  during any one minute.

                  If the ESME sends more messages (per minute)
                  than the maximum limit specified by 'ratelimit_minute',
                  these additional messages will be rejected by the
                  SMSC GW along with an error code - "throttled".

                  The default value for this parameter is "0" and it
                  implies "no restrictions". If this parameter is not
                  specified it implies "no restrictions".
```

```
ratelimit_hour    - This parameter is used to specify a maximum limit
                of messages that the SMSC will accept from this ESME
                during any one hour.

                If the ESME sends more messages (per hour)
                than the maximum limit specified by 'ratelimit_hour',
                these additional messages will be rejected by the
                SMSC GW along with an error code - "throttled".

                The default value for this parameter is "0" and it
                implies "no restrictions". If this parameter is not
                specified it implies "no restrictions".

ratelimit_day    - This parameter is used to specify a maximum limit
                of messages that the SMSC will accept from this ESME
                during any one day.

                If the ESME sends more messages (per day)
                than the maximum limit specified by 'ratelimit_day',
                these additional messages will be rejected by the
                SMSC GW along with an error code - "throttled".

                The default value for this parameter is "0" and it
                implies "no restrictions". If this parameter is not
                specified it implies "no restrictions".

min-message-length    - This paramter is used to specify the minimum
                message length (in characters) acceptable to
                the SMSC GW, for messages coming from this ESME.

                If an incoming message length is less than the
                min-message-length it will be rejected by SMSC GW.

                The default value for this parameter is "-1" and it
                implies "no limitations". Any other negative value
                also implies "no limitations".

max-message-length    - This paramter is used to specify the maximum
                message length (in characters) acceptable to
                the SMSC GW, for messages coming from this ESME.

                If an incoming message length is more than the
                max-message-length it will be rejected by SMSC GW.

                The default value for this parameter is "-1" and it
                implies "no limitations". Any other negative value
                also implies "no limitations".

enquire-server-enabled    - This parameter is used to enable or
                disable SMPP server sending enquire message.
```

```
                    The default value for this parameter is "false".


SEE ALSO
    smsc get scgt, smsc set scgt, smsc get scssn, smsc set scssn, smsc get hlrssn,
    smsc set hlrssn, smsc get mscssn, smsc set mscssn, smsc get maxmapv,
    smsc set maxmapv, smpp esme create
```

Using GUI

Procedure: Modify an existing ESME using GUI

1. In the GUI Management Console for SMSC Gateway, click on 'ESMEs' in the left panel.
2. The main panel will display the existing ESMEs (if any), one each in a row with corresponding actions (start, stop, delete, update) for each row.
3. You can update an existing by launching the 'ESME <name> properties' window by clicking on the blue coloured 'Update ESME' button. The 'ESME <name> properties' window will display all ESME paramters that can be updated by you. For more details of these parameters please refer to the descriptions of the CLI commands for the same in the preceding section.

   ESME can be setup for SSL so every connection request should first do SSL hand-shake. Settingup SSL is only possible from GUI. After creating the ESME, users can edit property and enable SSL.

   Only CLIENT ESME's (one that sends BIND request) can be enabled for SSL.

4. Update appropriate values for all the parameters and then click on the 'Close' button. This action will modify a new ESME with parameters as defined by you.
5. If there is an error in defining the ESME, then you will find the details of the error in the Management Console Log section below.

6.3.3. View ESME Details

Using CLI

You can view the details of all configured ESMEs by issuing the command `smpp esme show` as described below.

```
Name
    smpp esme show

SYNOPSIS
    smpp esme show

DESCRIPTION
    This command is used to list all configured ESMEs.
```

Using GUI

Procedure: View ESME using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'ESMEs' in the left panel.
2. The main panel will display the existing ESMEs (if any), one each in a row with corresponding actions (start, stop, delete) for each row.
3. You can view the details of an ESME by clicking on the row corresponding to the ESME. All relevant details of the ESME will be displayed in an expanded format.

6.3.4. Delete an existing ESME

Using CLI

You can delete any ESME by issuing the command `smpp esme delete` with appropriate parameters as described below.

```
Name
    smpp esme delete

SYNOPSIS
    smpp esme delete <esmeName>

DESCRIPTION
    This command is used to delete an existing ESME.

PARAMETERS
    esmeName        - Name of the ESME to be destroyed.
```

Using GUI

Procedure: Delete ESME using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'ESMEs' in the left panel.
2. The main panel will display the existing ESMEs (if any), one each in a row with corresponding actions (start, stop, delete) for each row.
3. To delete an existing ESME click on the delete icon marked 'x' in red, for the row corresponding to the ESME. You can delete an ESME only if it is stopped.

6.3.5. Start ESME

Using CLI

You can start an ESME by issuing the command `smpp esme start` with appropriate parameters as described below.

```
Name
    smpp esme start

SYNOPSIS
    smpp esme start <esmeName>

DESCRIPTION
    This command is used to start an existing ESME.

PARAMETERS
    esmeName        - Name of the ESME to be started.
```

Using GUI

Procedure: Start ESME using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'ESMEs' in the left panel.
2. The main panel will display the existing ESMEs (if any), one each in a row with corresponding actions (start, stop, delete) for each row.
3. To start an existing ESME click on the start icon lit in green, for the row corresponding to the ESME. You can start an ESME only if it is currently stopped.

6.3.6. Stop ESME

Using CLI

You can stop an ESME by issuing the command `smpp esme stop` with appropriate parameters as described below.

```
Name
    smpp esme stop

SYNOPSIS

DESCRIPTION
    This command is used to stop an already running ESME.

PARAMETERS
    esmeName        - Name of the ESME to be stopped.
```

Using GUI

Procedure: Stop ESME using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'ESMEs' in the left panel.
2. The main panel will display the existing ESMEs (if any), one each in a row with corresponding actions (start, stop, delete) for each row.

3. To stop an ESME click on the stop icon lit in red, for the row corresponding to the ESME. You can stop an ESME only if it is currently running.

6.3.7. Other ESME Operations

Using GUI

You can perform more operations in the GUI for any configured ESME. You can enable/disable Log Bytes and Log Pdu, dump window and reset counters.

Procedure: Other ESME Operations using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'ESMEs' in the left panel.
2. The main panel will display the existing ESMEs (if any), one each in a row with corresponding actions (start, stop, delete) for each row.
3. You can view the details of an ESME by clicking on the row corresponding to the ESME. All relevant details of the ESME will be displayed in an expanded format.
4. At the bottom of this expanded display you will find 6 buttons allowing you to perform the operations DisableLogBytes, DisableLogPdu, DumpWindow, EnableLogBytes, EnableLogPdu and ResetCounters.

6.4. SIP Settings

Restcomm SMSC comes with default JSLEE SIP RA. You can modify the SIP settings using the CLI or GUI.

6.4.1. Using CLI

You can modify the SIP settings by issuing the command `smsc sip modify` with appropriate parameters as described below. You can view the settings by issuing the command `smsc sip show` which will display all the values.

```
Name
    smsc sip modify

SYNOPSIS
    smsc sip modify name cluster-name <clusterName> host <ip> port <port>
    routing-ton <routing address ton> routing-npi <routing address npi>
    routing-range <routing address range> counters-enabled <true | false>
    charging-enabled <true | false> networkid <networkId>

DESCRIPTION
    This command is used to modify SIP settings.

PARAMETERS

    Standard Parameters
```

```
Name            - The name of the SIP Stack that is being modified. Since
                the Gateway does not allow creating more than one SIP
                Stack presently, the name is hardcored to "SIP".

Cluster-name        - The name of the Cluster to which this SIP Stack belongs
                to. This parameter is not used presently and is meant
                for future use when Cluster of SIP is enabled.

Host            - IP address of the remote node to which all SIP messages
                must be forwarded.

Port            - Port of the remote node to which all SIP messages
                must be forwarded.

routing-ton     - The DefaultSmsRoutingRule will try to match the
                'dest_addr_ton' of outgoing SMS with the value
                configured here. If this configured value is null(-1)
                or not null and matches, the SMSC will compare the
                'dest_addr_npi' and 'destination_addr' as explained
                below. If it doesn't match, the SMSC will select the
                next SIP in the list for matching routing rule.

routing-npi     - The DefaultSmsRoutingRule will try to match the
                'dest_addr_npi' of outgoing SMS with the value
                configured here. If this configured value is null(-1)
                or not null and matches, the SMSC will compare the
                'destination_addr' as below. If it doesn't match, the
                SMSC will select the next SIP in the list for matching
                routing rule.

routing-range       - The DefaultSmsRoutingRule will try to match the
                'destination_addr' of outgoing SMS with the value
                configured here. This is a regular java expression and
                default value is null. If it matches, the SMSC will
                send the SMS out over this SIP connection. If it
                doesn't match, the SMSC will select the next ESME in
                the list for matching routing rule.

counters-enabled    - Flag to enable or disable counters. Not used presently.

charging-enabled    - Flag to enable or disable charging for every SMS
                arriving from SIP.

networkId       - means to which virtual subnetwork belongs the SIP
                connector (this is for multi-tenancy support). Default value
                is 0. If you do not use multi-tenancy support - set this value
                to 0.
```

6.4.2. Using GUI

Procedure: Managing the SIP Connection using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'SIPs' in the left panel.
2. The main panel will display the existing SIP settings. You can view and modify the SIP settings in this panel.

6.5. MAP Version Cache

Restcomm SMSC caches the negotiated MAP Version for each network node. Cache stores the Global Title of SCCP address of the network node. Every MT SMS sent to this node will use the cached Version. This will help in reducing the MAP Version negotiation duration everytime and hence improve the performance.

6.5.1. Using CLI

You can set the 'MAP Version Cache' value by issuing the command `smsc mapcache set` with appropriate parameters as described below. You can verify this by issuing the command `smsc mapcache get` which will display all the cached values. Alternatively you can retrieve the cached value for a specific node by issuing the command `smsc mapcache get <node_digits>`. You can clear the cache by issuing the command `smsc mapcache clear`.

```
Name
    smsc mapcache set

SYNOPSIS
    smsc mapcache set <node_digits> <version>

DESCRIPTION
    This command is used to set the version for the specific node digits. SMSC Gateway
    caches the negotiated MAP Version for each network node. Every MT SMS sent to this
    node will use the cached Version.
```

6.5.2. Using GUI

Procedure: Managing MAP Version Cache value using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'MAP Version Cache' in the left panel.
2. You can specify the MAP Application Context version for every node by entering the value for Node Digits and choosing the Version for MAP MtForwardSm operation.
3. You must click on the button 'Update' to save your settings. If there is an error in setting the value, then you will find the details of the error in the Management Console Log section below.
4. To view the cached version value for a specific node, enter the node digits (under View Cache) and click on the "View" button. The cached value will be displayed in the log.
5. To clear all the cached values, click on the "Clear Cache" button.

6.6. Database Routing Rules

Restcomm SMSC can now act as a SMPP Server accepting incoming SMPP connections or can also act as a SMPP client initiating a connection to a remote SMSC server. Therefore you can now set an intelligent database routing rule to route SMS between SMPP connections or between SMPP, SIP and GSM.

By default, Restcomm SMSC is setup to leverage the "routing address-range" to decide the routing of SMS. For example:

1) If Server ESME is defined of type RECEIVER (and hence accepts incoming RECEIVER BIND from peer client) with routing address range as 6666; if SMS arrives in system destined for 6666, it matches with this ESME and SMS will be sent as DELIVER_SM to client side.

2) If Client ESME is defined of type TRANSMITTER (and hence initiates TRANSMITTER BIND to peer server) with routing address range as 6666; if SMS arrives in system destined for 6666, it matches with this ESME and SMS will be sent as SUBMIT_SM to server side.

The destined number of SMS is checked with address range. Restcomm SMSC uses regular expression for matching the pattern.

The above methodology can work if the routing rule is based on MSISDN range, however if the range is not fixed (like in case of Number Portability) the above process will break. In such cases you can define a Database routing rule. You must change the value of the property `smsRoutingRuleClass` in the file *RestComm-smscgateway-<version>/jboss-5.1.0.GA/server/default/deploy/restcomm-smsc-server/META-INF/jboss-beans.xml* to look like below and un-comment if its commented out:

```
<property
name="smsRoutingRuleClass">org.mobicents.smsc.smpp.DatabaseSmsRoutingRule</property>
```

SMSC stores the routing rule in the Cassandra Database. You can populate this table with address and corresponding cluster name as explained in the sections below.

6.6.1. Create/Update Database Routing Rule

Using CLI

You can create or update a Database Routing Rule using the command `smsc databaserule update` with appropriate parameters as described below:

```
Name
    smsc databaserule update

SYNOPSIS
    smsc databaserule update <address> <systemId> <SMPP|SIP> networkid <networkId>

DESCRIPTION
    This command is used to add or update a Database Rule for SMPP or SIP.

    The parameter <SMPP|SIP> is used to define if the rule is for SMPP or SIP. This
    is an optional parameter and if unspecified, by default the rule is set for SMPP.

    Database Rules are Rules that are used for routing messages to a proper ESME.
    When you define a rule using the above command, you are creating a routing rule
    that states:
    "If the destination address of a message corresponds with the value specified in
    the 'address' field, then the message be sent to an ESME identified by the value
    specified in the 'systemId' field".

    To add a new rule you must issue the command with the systemId parameter and
    specify if the rule is for SIP or SMPP.

    To update an existing rule, you must issue the command with both the address
    parameter and the systemId parameter and specifiy if the rule is for SIP or SMPP.

    networkId - means to which virtual SS7 subnetwork belongs a database
                routing rule (this is for Multi-tenancy support). If this
                parameter is skipped - networkId will be set to "0" for
                a database routing rule operation.
```

Using GUI

Procedure: Create/Update Database Routing Rule using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'DB Routing Rule' in the left panel.
2. The main panel will allow you to create/update, delete and view DB Routing Rules for SMPP or SIP.
3. In order to create or update a DB Routing Rule, choose the type as SMPP or SIP from the drop down box, enter the values for MSISDN and ESME cluster name and click on 'Update'. A new rule will be created if it does not exist or updated if it exists.

6.6.2. Delete Database Routing Rule

Using CLI

You can delete a Database Routing Rule using the command `smsc databaserule delete` with appropriate parameters as described below:

```
Name
    smsc databaserule delete

SYNOPSIS
    smsc databaserule delete <address> <SMPP|SIP> networkid <networkId>

DESCRIPTION
    This command is used to delete an existing Database Rule specified for 'address'.

    The parameter <SMPP|SIP> is used to define if the rule is deleted for SMPP or SIP.
    This is an optional parameter and if unspecified, by default the rule is
    deleted for SMPP.

    networkId - means to which virtual SS7 subnetwork belongs a database
                routing rule (this is for Multi-tenancy support). If this
                parameter is skipped - networkId will be set to "0" for
                a database routing rule operation.
```

Using GUI

Procedure: Delete Database Routing Rule using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'DB Routing Rule' in the left panel.
2. The main panel will allow you to create/update, delete and view DB Routing Rules for SMPP or SIP.
3. In order to delete a DB Routing Rule, choose the type as SMPP or SIP from the drop down box, enter the value for MSISDN and click on 'Delete'. The routing rule corresponding to that MSISDN will be deleted.

6.6.3. View Database Routing Rule Information

Using CLI

You can view a Database Routing Rule using the command `smsc databaserule get` with appropriate parameters as described below:

```
Name
    smsc databaserule get

SYNOPSIS
    smsc databaserule get <address> <SMPP|SIP> networkid <networkId>

DESCRIPTION
    This command is used to view the details of an existing Database Rule specified
    for 'address'.

    The parameter <SMPP|SIP> is used to define if the rule is to be viewed for SMPP
    or SIP. This is an optional parameter and if unspecified, by default the rule is
    retrieved for SMPP.

    networkId - means to which virtual SS7 subnetwork belongs a database
                routing rule (this is for Multi-tenancy support). If this
                parameter is skipped - networkId will be set to "0" for
                a database routing rule operation.
```

Using GUI

Procedure: View Database Routing Rule using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'DB Routing Rule' in the left panel.
2. The main panel will allow you to create/update, delete and view DB Routing Rules for SMPP or SIP.
3. In order to view a DB Routing Rule, choose the type as SMPP or SIP from the drop down box, enter the value for MSISDN and click on 'View'. The routing rule corresponding to that MSISDN will be displayed.

6.6.4. Retrieve a range of Database Routing Rules

Using CLI

You can retrieve a range of Database Routing Rules using the command `smsc databaserule getrange` with appropriate parameters as described below:

```
Name
    smsc databaserule getrange

SYNOPSIS
    smsc databaserule getrange <SMPP|SIP> <address>

DESCRIPTION
    This command is used to retrieve a list of database rules as text data.

PARAMETERS
    Standard Parameters:
    <SMPP|SIP> -  This parameter is used to specify if you wish to retrieve the
            range corresponding to SMPP or SIP.

    Optional Parameters:
    <address> -   If a value is not specified for <address>, then the command will
            retrieve the first 100 database rules.

            If <address> is specified, then the command will retrieve a list
            of 100 database rules starting from the record next to the record
            with address='address'.
```

6.7. Message processing rules (mproc rules)

The fundamentals of mproc rules you can in the chapter Message processing rules (mproc rules). Here we will describe how to manage mproc rules.

6.7.1. Create a mproc rule.

Using CLI

You can configure a new mproc rule by issuing the command smsc mproc add with appropriate parameters as described below.

```
Name
    smsc mproc add

SYNOPSIS
    smsc mproc add <class name> <id> desttonmask <destination type of number>
    destnpimask <destination numbering plan indicator> destdigmask <regular
    expression - destination number digits mask> originatingmask <SS7_MO |
    SS7_HR | SMPP | SIP> originatorsccpaddressmask <originator SCCP
    CallingPartyAddress mask> networkidmask <networkId value> origesmenamemask
    <regualr expression - origination ESME name mask> newnetworkid
    <new networkId value> newdestton<new destination type of number> newdestnpi
    <new destination numbering plan indicator> adddestdigprefix <prefix>
    makecopy <false | true> dropaftersri <false | true>

DESCRIPTION

    This command is used to add a new mproc rule.

PARAMETERS

    Standard Parameters.
    class name - the name of class of mproc rules implementation. For the
      default implementation (that is described here) class name is "mproc".
    id - a mandatory parameter that means a unique mproc identifier.
    desttonmask - an optional parameter.
    destnpimask - an optional parameter.
    destdigmask - an optional parameter.
    originatingmask - an optional parameter.
    originatorsccpaddressmask - an optional parameter.
    networkidmask - an optional parameter.
    origesmenamemask - an optional parameter.
    newnetworkid - an optional parameter.
    newdestton - an optional parameter.
    newdestnpi - an optional parameter.
    adddestdigprefix - an optional parameter.
    makecopy - an optional parameter.
    dropaftersri - an optional parameter.
    Description of parameters is in the previous chapter.

EXAMPLES
    smsc mproc add mproc 1 networkidmask 11 newnetworkid 12
```

Using GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Message Processing Rules' in the left panel.
2. The main panel will display the existing mproc rule (if any), one each in a row with corresponding actions (delete, update) for each row. Below this you will find buttons 'Create Default Message Processing Rule' and 'Create Custom Message Processing Rule'. If you have not

implemented your own customized rule, the only option for you is 'Create Default Message Processing Rule'. The 'Create Message Processing Rule' window will display all proc rule parameters that must be defined by you. For more details of these parameters please refer to the descriptions of the CLI commands for the same in the preceding section.

3. If you have implemented your own customized rules than you can use the button 'Create Custom Message Processing Rule'. For a custom mproc rule you need to configure rule Id, Class Name and Parameters (as a plain string that will be parsed by your customized rules).
4. Enter appropriate values for all the parameters and then click on the 'Create' button at the bottom of this 'Create Message Processing Rule' window. This action will create a new mproc rule with parameters as defined by you.
5. If there is an error in defining the mproc rule, then you will find the details of the error in the Management Console Log section below.

6.7.2. Modify a mproc rule.

Using CLI

You can modify an existent mproc rule by issuing the command smsc mproc modify with appropriate parameters as described below.

```
Name
    smsc mproc modify

SYNOPSIS
    smsc mproc modify <id> desttonmask <destination type of number> destnpimask
    <destination numbering plan indicator> destdigmask <regular expression -
    destination number digits mask> originatingmask <SS7_MO | SS7_HR | SMPP
    | SIP> originatorsccpaddressmask <originator SCCP CallingPartyAddress mask>
    networkidmask <networkId value> origesmenamemask <regualr expression
    - origination ESME name mask> newnetworkid <new networkId value>
    newdestton<new destination type of number> newdestnpi <new destination
    numbering plan indicator> adddestdigprefix <prefix> makecopy <false | true>
    dropaftersri <false | true>

DESCRIPTION

    This command is used to modify an existent mproc rule.

PARAMETERS

    Standard Parameters.
    id - a mandatory parameter that means a unique mproc identifier.
    desttonmask - an optional parameter.
    destnpimask - an optional parameter.
    destdigmask - an optional parameter.
    originatingmask - an optional parameter.
    originatorsccpaddressmask - an optional parameter.
    networkidmask - an optional parameter.
    origesmenamemask - an optional parameter.
    newnetworkid - an optional parameter.
    newdestton - an optional parameter.
    newdestnpi - an optional parameter.
    adddestdigprefix - an optional parameter.
    makecopy - an optional parameter.
    dropaftersri - an optional parameter.

EXAMPLES
    smsc mproc modify 1 newnetworkid 13
```

Using GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Message Processing Rules' in the left panel.
2. The main panel will display the existing mproc rule (if any), one each in a row with corresponding actions (delete, update) for each row. Below this you will find the button 'Create Message Processing Rule'.
3. You can modify an existent mproc rule by launching the 'Message Procesing Rule #... properties' window by clicking on the blue coloured 'Modify Message Processing Rule' button. The 'Message

Procesing Rule #... properties' window will display all proc rule paramters that must be updated by you. For more details of these parameters please refer to the descriptions of the CLI commands for the same in the preceding section. For customized mproc rules the set of parameters is configured as a plain string.

4. Update appropriate values for all the parameters and then click on the 'Close' button. This action will modify a mproc rule with parameters as defined by you.
5. If there is an error in updating the mproc rule, then you will find the details of the error in the Management Console Log section below.

6.7.3. View a mproc rule details.

Using CLI

You can view the details of all configured mproc rules or a specified mproc rule by issuing the command smsc mproc show as described below.

```
Name
    smsc mproc show

SYNOPSIS
    smsc mproc show <id>

DESCRIPTION

    This command is used to list all configured mproc rules
    or a specified mproc rule. Only nondefault mproc rule parameters
    (conditions and actions) will be displayed in the command output.

PARAMETERS

    id - an optional parameter. You can specify this parameter to ask
    detalis for a mproc rule with a provided Id. If you do not
    specify this parameter all mproc rules will be displayed.

EXAMPLES
    smsc mproc show 1
    smsc mproc show
```

Using GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Message Processing Rules' in the left panel.
2. The main panel will display the existing mproc rule (if any), one each in a row with corresponding actions (delete, update) for each row. Below this you will find the button 'Create Message Processing Rule'.
3. You can view the details of a mproc rule by clicking on the row corresponding to the mproc rule. All relevant details of the mproc rule will be displayed in an expanded format.

6.7.4. Remove an existing mproc rule.

Using CLI

You can remove an existent mproc rule by issuing the command smsc mproc remove with appropriate parameters as described below.

```
Name
    smsc mproc remove

SYNOPSIS
    smsc mproc remove <id>

DESCRIPTION

    This command is used to remove an existing mproc rule.

PARAMETERS

    id - a mandatory parameter - id of an existent
    mproc rule to remove.

EXAMPLES
    smsc mproc remove 1
```

Using GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Message Processing Rules' in the left panel.
2. The main panel will display the existing mproc rule (if any), one each in a row with corresponding actions (delete, update) for each row. Below this you will find the button 'Create Message Processing Rule'.
3. To remove an existing mproc rule click on the delete icon marked 'x' in red, for the row corresponding to the mproc rule.

6.8. Statistics

The GUI will allow you to create campaigns of fixed duration for gathering statistics data. Campaign allows to select time period over which these statistics have been gathered (in hours, minutes and seconds). Once Campaign is defined, the statistics can be observed by clicking on the newly created campaign name or you can also navigate to Metrics (click Metrics on left panel) to get graph of statistics.

6.8.1. Create new Campaign

Using GUI

Procedure: Create new Campaign using GUI

1. To create a new Campaign open a Web Browser and navigate to http://localhost:8080/jss7-management-console/. Click on the 'Manage Campaigns' link in the left panel. The main panel will display the names of all existing campaigns and also a button to create a new campaign. The GUI will look similar to the figure below.

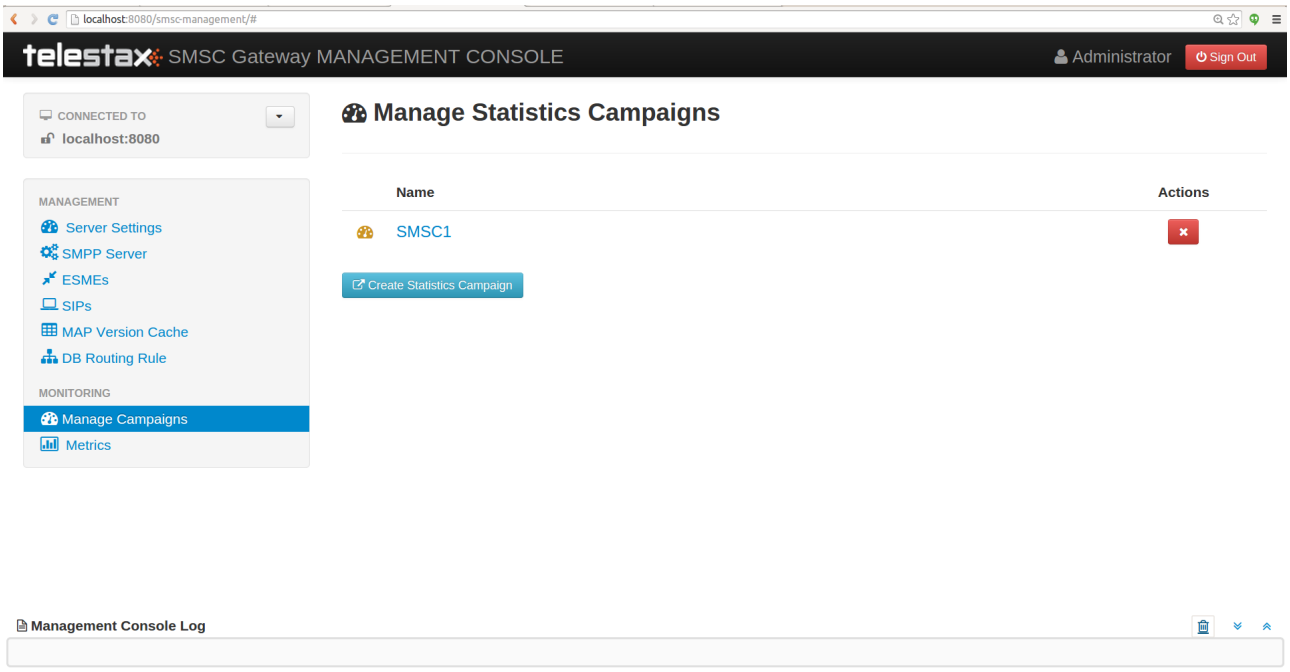


Figure 4. GUI - Campaigns

2. Click on the 'Create Statistics Campaign' button to create a new Campaign. Select the stack from the drop down 'Counter Definition Set Name' on which you want the new campaign to be defined.
3. Select the time period from the drop down 'Duration' and enter a unique 'Campaign Name'.

The drop down will also display SS7 counter definiton. You can create SS7 campaigns from the SS7 managment console. For SMSC, select 'SMSC GW-SMSC-Main'.

6.8.2. View Campaigns

Using GUI

You can view all existing campaigns in the GUI. On the main panel, click on the Campaign name. The GUI will look similar to the figure below and is divided into tabs. The first tab will display the properties of the campaign. The second tab explains all the counters in this campaigns and their definition. The last tab provides the values for each of these counters. The last tab also displays the 'Start Time' and 'End Time' representing the time duration for which the sample was collected.

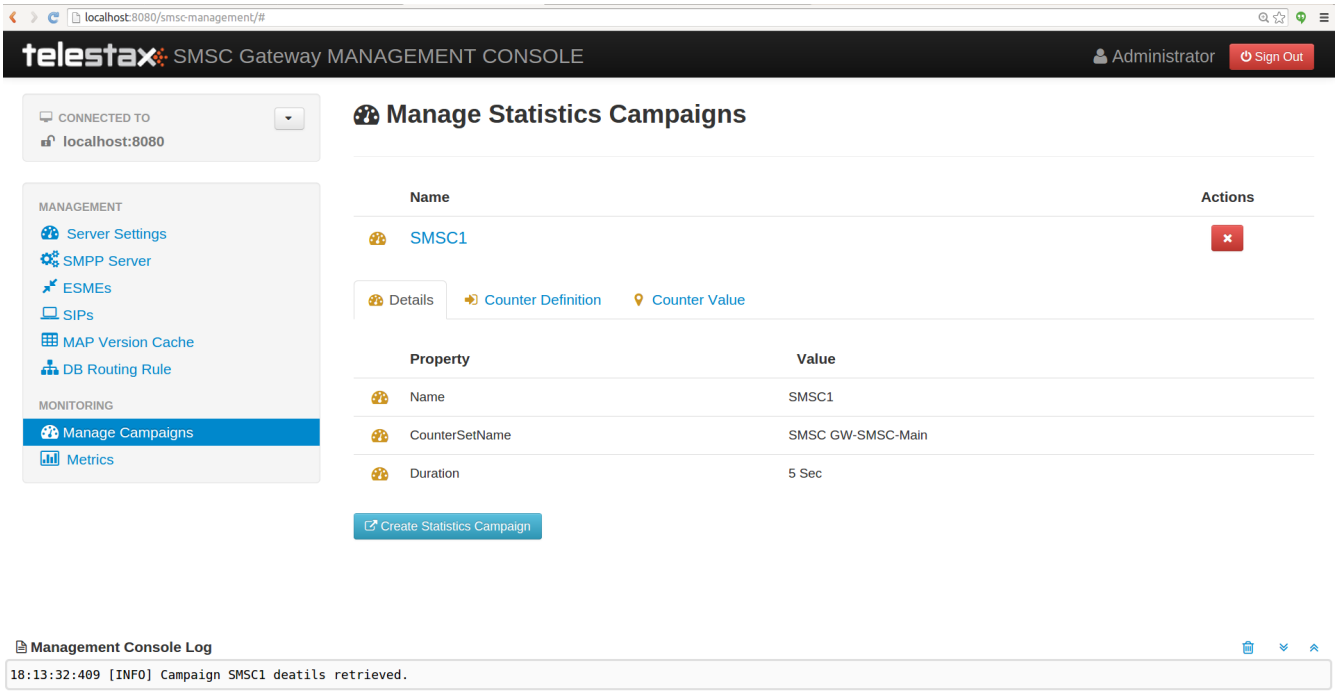


Figure 5. GUI - Campaigns View

Restcomm SMSC doesn't persist the statistics, hence the data collected for the campaign period refreshes for every defined 'Duration'. You must refresh the page for every 'Duration' period to gather statistics data for the previous time period.

Nevertheless you can also click on the 'Metrics' link on left panel, select the Campaign and observe the statistics graph. The metrics page gathers data from the time the page was loaded till user navigates away. Hence graph will show historic data from the point the page was loaded.

Metrics will show 3 graphs for Messages coming in, Messages attempted for delivery and Messages successfully delivered.

Figure 6. GUI - Campaigns View

Chapter 7. Maintenance

7.1. Database

The Restcomm SMSC creates three tables for every new day to store data relevant to the messages scheduled for delivery on that day, as explained in [_database_table_structure]. These tables need to be dropped periodically else the tables may end up occupying huge disk space.

You can choose to drop these tables manually or configure the Gateway to drop the tables periodically. To configure these settings, you can make use of the parameters `removinglivetablesdays` and `removingarchivetabledays`.

By default, Cassandra Database creates snapshots of tables prior to dropping them. Therefore the disk space is not cleared upon dropping the tables. To overcome this, you must update the *cassandra.yaml* configuration file and set the value of `auto_snapshot` to false.

7.1.1. Automatically drop tables

Using CLI

You can configure the auto-drop settings by issuing the command `smsc set removinglivetablesdays` for LIVE tables and the command `smsc set removingarchivetabledays`, with appropriate parameters as described below. You can verify this by issuing the command `smsc get removinglivetablesdays` and `smsc get removingarchivetabledays` which will display the values set for these properties.

```
Name
    smsc set removinglivetablesdays

SYNOPSIS
    smsc set removinglivetablesdays <value>

DESCRIPTION
    This command is used to configure the SMSC to automatically drop LIVE tables from
    the Cassandra Database. The SMSC will attempt to delete tables just after
    midnight and after every SMSC restart.

PARAMETERS
    removinglivetablesdays      - This parameter is used to specify the number
                      of days the LIVE tables should be kept before
                      attempting to drop them automatically.

                      If this value is specified as "0", the SMSC
                      will not drop tables automatically. In this case
                      you must manually drop tables.

                      You must specify a value of 3 or more. You can
                      not set this value to 1 or 2 days. This is to
                      ensure the tables will be kept for a minimum
                      of 2 days after creation date.

                      The SMSC wil attempt to delete tables for one
                      day. If the Cassandra Database keeps tables for
                      older days, then the administrator should drop
                      these manually.
```

```
Name
    smsc set removingarchivetabledays

SYNOPSIS
    smsc set removingarchivetabledays <value>

DESCRIPTION
    This command is used to configure the SMSC to automatically drop ARCHIVE tables
    from the Cassandra Database. The SMSC will attempt to delete tables just after
    midnight and after every SMSC restart.

PARAMETERS
    removingarchivetabledays    - This parameter is used to specify the number
                      of days the ARCHIVE tables should be kept before
                      attempting to drop them automatically.

                      If this value is specified as "0", the SMSC
                      will not drop tables automatically. In this case
                      you must manually drop tables.

                      You must specify a value of 3 or more. You can
                      not set this value to 1 or 2 days. This is to
                      ensure the tables will be kept for a minimum
                      of 2 days after creation date.

                      The SMSC wil attempt to delete tables for one
                      day. If the Cassandra Database keeps tables for
                      older days, then the administrator should drop
                      these manually.
```

7.1.2. Manually drop tables

If you do not want the SMSC Gateway to automatically drop tables you must set the value of the parameters `removinglivetablesdays` and `removingarchivetabledays` to zero and disable this feature as explained in the preceding section.

You must decide on how much data you would like to retain in the Database and accordingly delete the tables. However you must take precaution to not delete the tables for today or a future date. All tables that store data for dates less than the current date may be safely deleted from the database without stopping the SMSC.

To delete the tables manually you must run the required commands via Cassandra CQL3 as in the examples below:

```
DROP TABLE DST_SLOT_TABLE_2014_01_05;
DROP TABLE SLOT_MESSAGES_TABLE_2014_01_05;
DROP TABLE MESSAGES_2014_01_05;
```

Chapter 8. Monitoring

8.1. View SMSC Statistics

8.1.1. Using CLI

You can view the current state of SMSC using the command `smsc stat get` with appropriate parameters as described below:

```
Name
    smsc stat get

SYNOPSIS
    smsc stat get

DESCRIPTION
    This command is used to view the details of the current state of the SMSC and
    monitor the SMSC. The output prints the following parameters:

    Time            - Current time. By obtaining this statistic twice, this
                  value can be used to calculate the time interval between
                  statistic time.

    MessageInProcess    - Number of messages currently being processed for
                  delivery in GSM (MT messages) and SMPP (messages that
                  are routed to ESME) and SIP (messages that are routed to
                  ESME).

    MessageId       - This is the last assigned Message Id. This indicates
                  the number of messages that have come into the SMSC
                  from GSM (Mobile Originated messages), from ESMEs or from SIP
                  and stored in the Cassandra database. The MessageId
                  counter is initiated from the time of the installation
                  of the SMSC.

    MessageScheduledTotal   - The number of messages put into the delivering
                  process in both GSM, SMPP or SIP since the SMSC was started.

    DueSlotProcessingLag    - The time (in seconds) between "in time" due_slot
                  and "in process" due_slot. "In time" means the current actual
                  time, "in process" means that SMSC GW is processing messages
                  that was scheduled for that time (in the past).

                  If this value is equal to 0 or 1 or 2, it means that the
                  SMSC is not highly loaded and all the messages are
                  being processed on time.

                  If this value is high, say for example 300, then it
                  means the SMSC is overloaded and is now processing
```

```
                messages, that are scheduled for processing 300 seconds
                before the current time.

                If this value is progressively increasing, then the SMSC
                is heavily overloaded and the incoming messages count at
                SMPP (and MO) are more than what the SMSC can deliver.

                If the incoming messages are not many, this value will
                decrease and will reach 0 when there are no messages.

                Normally this value will return to 0 except for few
                peaks. If it does not, then you must reduce the load for
                SMSC.

    DueSlotProcessingTime - This field shows the time for which SMSC GW is
                processing / fetching stored in cassandra database messages
                ("in process" due_slot). If this time is far before the
                current actual time, then this means:
                - either SMSC GW is overloaded and can not send messages
                  in time
                - or SMSC GW was turned off for much time and now is
                  checking for messages for the time when it was off
                If you want to skip unsent messages that was scheduled for
                the time in the past (and then shift DueSlotProcessingTime
                to to current time) - you can use the command [the reference
                to CLI / GUI command "smsc skip-unsent-messages" - see the
                Chapter "Skipping of scheduled for the past and not yet
                sent messages ("In process" due_slot shifting)."

    Param1          - Ignore.

    Param2          - Ignore.

    SmscStartTime       - The time when the SMSC was started.
EXAMPLES
    smsc stat get
    Stat: Time: Mon Jan 13 18:40:42 CET 2014, MessageInProcess: 515, MessageId: 10212,
    MessageScheduledTotal: 8992, DueSlotProcessingLag: 0, Param1: 0, Param2: 0,
    SmscStartTime: Mon Jan 13 18:39:30 CET 2014
```

8.1.2. Using GUI

Procedure: View SMSC Statistics using the GUI

1. In the GUI Management Console for SMSC Gateway, click on 'Stats' in the left panel.
2. The main panel will display the current statistics of the SMSC and display the details of the parameters 'START TIME', 'CURRENT TIME', 'TOTAL MESSAGES SCHEDULED', 'MESSAGES IN PROCESS', 'CURRENT MESSAGE ID' and 'DUE SLOT PROCESSING LAG'. For more details of these parameters please refer to the description of the CLI command in the preceding section.

This page gets auto-refreshed every 2.5 seconds and therefore the statistics get refreshed automatically.

8.2. CDR Log

Restcomm SMSC is configured to generate CDR in a plain text file located at *restcomm-smscgateway-<version>/jboss-5.1.0.GA/server/<profile>/log/cdr.log*. The CDR generated in the text file is of the below format:

```
SIMBIT_DATE,ADDR_SRC_DIGITS,ADDR_SRC_TON,ADDR_SRC_NPI,ADDR_DST_DIGITS,ADDR_DST_TON,ADD
R_DST_NPI,Message_Delivery_Status,ORIG_SYSTEM_ID,MESSAGE_ID,NNN_DIGITS,IMSI,CORR_ID,Fi
rst 20 characters of SMS, Reason_For_Failure
```

where `ADDR_SRC_DIGITS`, `ADDR_SRC_TON`, `ADDR_SRC_NPI`, `ADDR_DST_DIGITS`, `ADDR_DST_TON`, `ADDR_DST_NPI`, `ORIG_SYSTEM_ID`, `MESSAGE_ID`, `NNN_DIGITS`, `IMSI` and `CORR_ID` are as explained in [_slot_messages_table_yyyy_mm_dd]; `SIMBIT_DATE`, `Message_Delivery_Status` and `Reason_For_Failure` are described below in this section.

`NNN_DIGITS` and `IMSI` fields are present only in the case of SS7 terminated messages when there is a SRI positive response. `CORR_ID` is present only if a message has come to the SMSC Gateway via "home-routing" procedure.

SIMBIT_DATE

Time when the message reached the SMSC Gateway.

Message_Delivery_Status

The CDR text file contains a special field, `Message_Delivery_Status`, that specifies the message delivery status. The possible values are described below:

Message_Delivery_Status if delivering to GSM network:

partial

Delivered a part of a multi-part message but not the last part.

success

Delivered the last part of a multi-part message or a single message.

temp_failed

Failed delivering a part of a multi-part message or a single message. It does not indicate if a resend will be attempted or not.

failed

Failed delivering a message and the SMSC will now attempt to resend the message or part of the message.

failed_imsi

Delivery process was broken by a mproc rule applying at the step when a successful SRI

response has been received from HLR.

Message_Delivery_Status if delivering to ESME:

partial_esme

Delivered a part of a multi-part message but not the last part.

success_esme

Delivered the last part of a multi-part message or a single message.

temp_failed_esme

Failed delivering a part of a multi-part message or a single message.

failed_esme

Failed delivering a message and the SMSC will now attempt to resend the message or part of the message.

Message_Delivery_Status if delivering to SIP:

partial_sip

Delivered a part of a multi-part message but not the last part.

success_sip

Delivered the last part of a multi-part message or a single message.

temp_failed_sip

Failed delivering a part of a multi-part message or a single message.

failed_sip

Failed delivering a message and the SMSC will now attempt to resend the message or part of the message.

Message_Delivery_Status if the message has been rejected by the OCS Server (Diameter Server):

ocs_rejected

OCS Server rejected an incoming message.

Message_Delivery_Status if the message has been rejected by a mproc rule applying at the step when a message has been arrived to SMSC GW:

mproc_rejected

A mproc rule rejected an incoming message (and reject response was sent to a message originator).

mproc_dropped

A mproc rule dropped an incoming message (and accept response was sent to a message originator).

. The last field in the CDR generated is `Reason_For_Failure`, which records the reason for delivery failure and is empty if the delivery is successful. The possible delivery failure cases are explained below.

Reasons_For_Failure

XXX response from HLR

A MAP error message is received from HLR after SRI request; XXX: `AbsentSubscriber`, `AbsentSubscriberSM`, `CallBarred`, `FacilityNotSuppored`, `SystemFailure`, `UnknownSubscriber`, `DataMissing`, `UnexpectedDataValue`, `TeleserviceNotProvisioned`.

Error response from HLR: xxx

Another MAP error message is received from HLR after SRI request.

Error XXX after `MtForwardSM` *Request*

A MAP error message is received from MSC/VLR after `MtForwardSM` request; XXX: `subscriberBusyForMtSms`, `absentSubscriber`, `absentSubscriberSM`, `smDeliveryFailure`, `systemFailure`, `facilityNotSup`, `dataMissing`, `unexpectedDataValue`, `facilityNotSupported`, `unidentifiedSubscriber`, `illegalSubscriber`.

Error after `MtForwardSM` *Request: xxx*

Another MAP error message is received from MSC/VLR after `MtForwardSM` request.

DialogClose after `MtRequest`

No `MtForwardSM` response and no error message received after `MtForwardSM` request.

`onDialogProviderAbort` *after* `MtForwardSM` *Request*

MAP `DialogProviderAbort` is received after `MtForwardSM` request.

`onDialogProviderAbort` *after SRI Request*

MAP `DialogProviderAbort` is received after SRI request.

Error condition when invoking `sendMtSms()` *from* `onDialogReject()`

After a `MtForwardSM` request MAP version conflict, MAP message negotiation was processed but this process failed, or other fundamental MAP error occurred.

`onDialogReject` *after SRI Request*

After a SRI request MAP version conflict, MAP message negotiation was processed but this process failed, or other fundamental MAP error occurred.

`onDialogTimeout` *after* `MtForwardSM` *Request*

Dialog timeout occurred after `MtForwardSM` Request. The reason may be GSM network connection failure or SMSC overload.

`onDialogTimeout` *after SRI Request*

Dialog timeout occurred after SRI Request. The reason may be GSM network connection failure or SMSC overload.

`onDialogUserAbort` *after* `MtForwardSM` *Request*

`DialogUserAbort` message is received from a peer or sent to a peer. The reason may be GSM fundamental failure or SMSC overload.

`onDialogUserAbort` *after SRI Request*

`DialogUserAbort` message is received from a peer or sent to a peer. The reason may be GSM fundamental failure or SMSC overload.

`onRejectComponent` *after* `MtForwardSM` *Request*

Reject component was received from a peer or sent to a peer. This is an abnormal case and implies MAP incompatibility.

`onRejectComponent` *after SRI Request*

Reject component was received from a peer or sent to a peer. This is an abnormal case and implies MAP incompatibility.

Other

Any other message that usually indicates some internal failure.

Chapter 9. SMSC RestComm-Connect Integration

RestComm-Connect is a next generation cloud communications platform to rapidly build voice, video and realtime messaging applications, using mainstream development skills.

RestComm-Connect is a turnkey Cloud Communications platform based on Open Source building blocks from the team behind Restcomm. RestComm offers a clean room implementation of the Twilio.com APIs and much more.

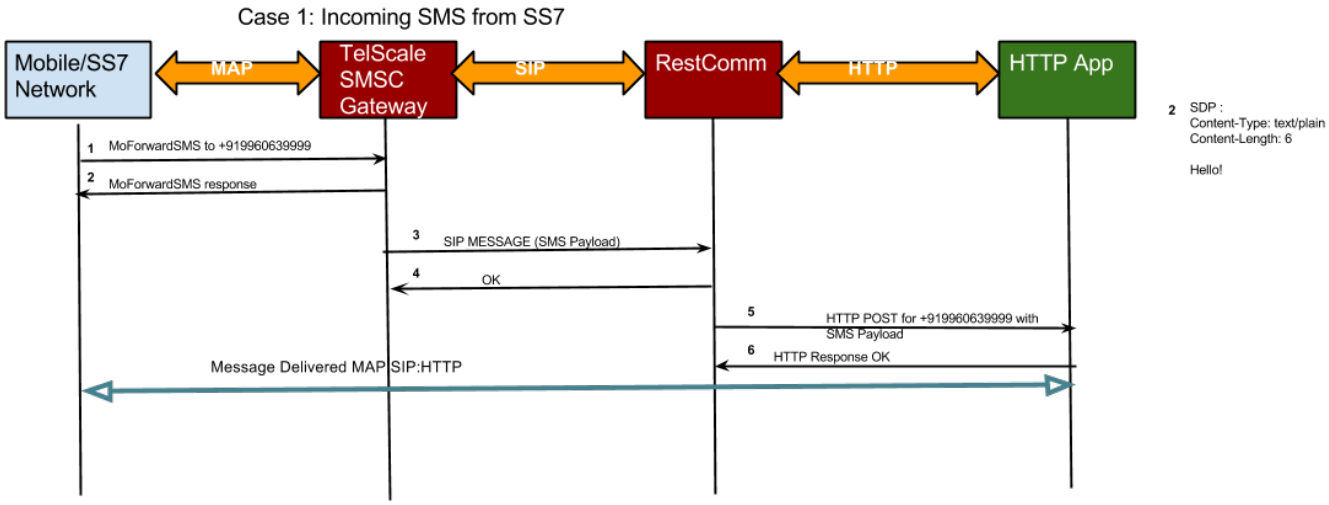
RestComm-Connect can be integrated with VoIP and legacy SS7 network providers either in the cloud or via on-premise Resource Adaptors. For further details on RestComm please visit restcomm.com

9.1. Restcomm SMSC Integration with RestComm-Connect Architecture

Restcomm SMSC integrates with RestComm-Connect over SIP. The below sections describe flow of SMS messages between SMSC, RestComm and Applications over RestComm.

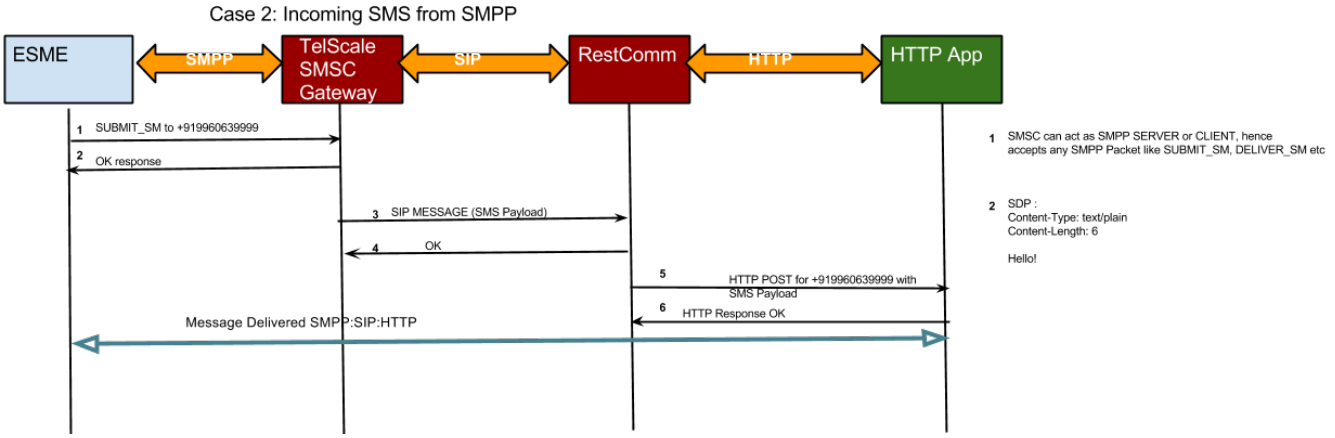
9.1.1. Incoming SS7 message

Below diagram shows flow of a message coming from SS7 and getting delivered to Application over RestComm-Connect.



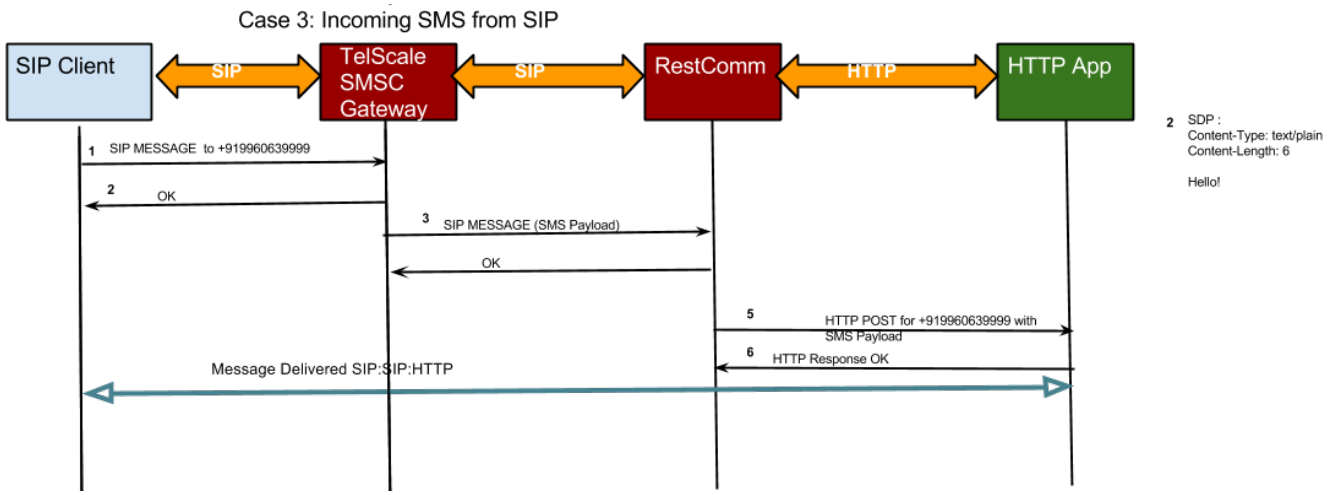
9.1.2. Incoming SMPP message

Below diagram shows flow of a message coming from SMPP and getting delivered to Application over RestComm-Connect.



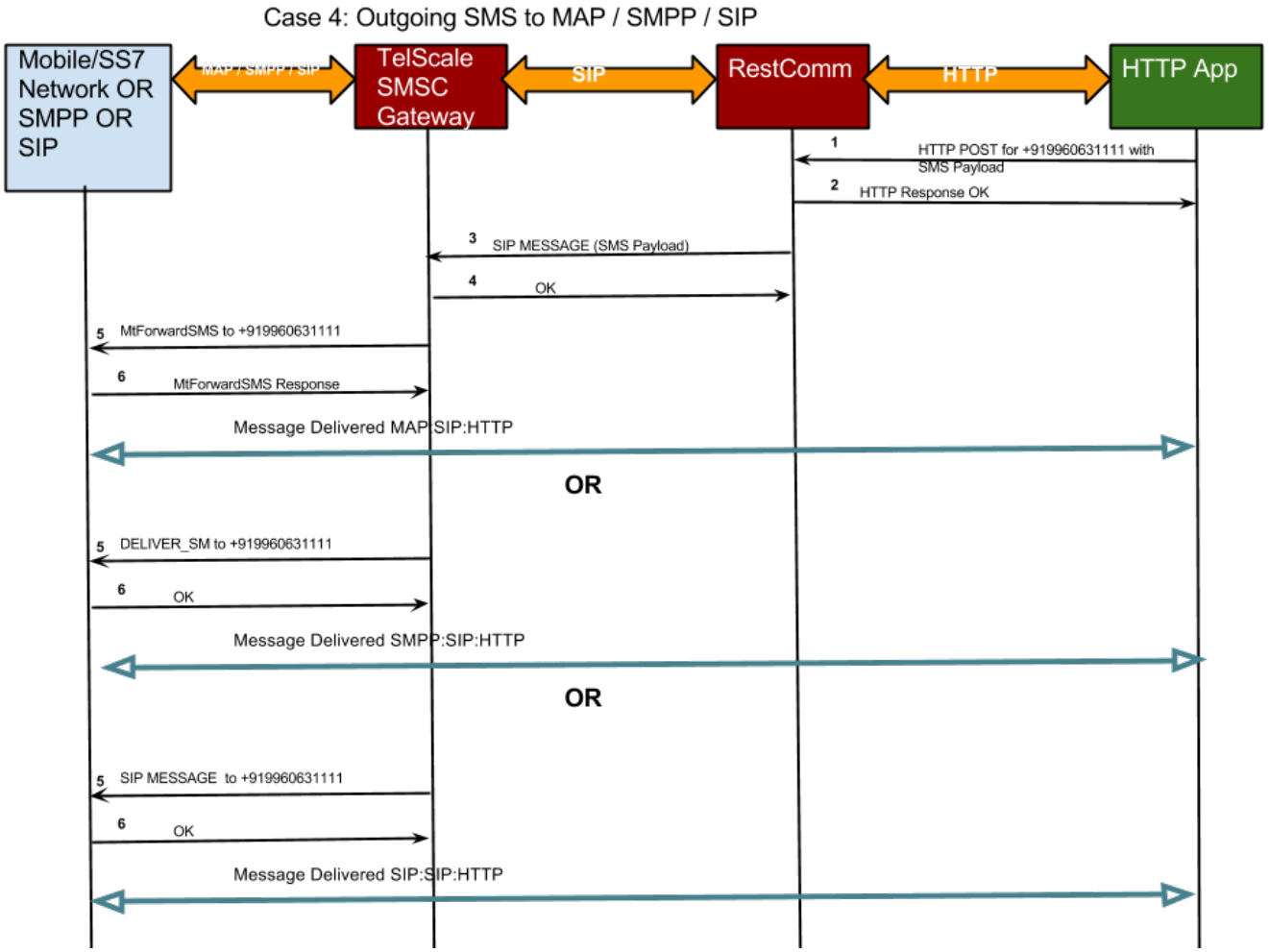
9.1.3. Incoming SIP message

Below diagram shows flow of a message coming from SIP and getting delivered to Application over RestComm-Connect.



9.1.4. Outgoing message from Application over RestComm

Below diagram shows flow of a message from application over RestComm-Connect and getting delivered to Restcomm SMSC .



9.2. Customized SIP Headers

SMS messages can be of UTF-8 or Unicode Characters, or there can be special messages with user data headers (UDH), binary messages, SMS that indicates delivery receipt etc. There is no one-to-one mapping for this information in SIP, hence Restcomm RestComm has introduced additional customized SIP Headers to convey this useful information as described below.

To understand each of the below optional parameters please read SMPP Specifictaion.

X-SMS-UDH

This parameter carries the hex string of User Data Header. For example, X-SMS-UDH=06050413011301. RestComm can include this header for outgoing SIP MESSAGE and SMSC will include it in actual SMS. If incoming SMS has UDH, then the SIP MESSAGE from SMSC will include this optional header.

X-SMS-CODING

This is the parameter for message coding. When unset, defaults to 0 (7 bits) if Content-Type is text/plain , text/html or text/vnd.wap.wml; for application/octet-stream, defaults to 8 bits (1); 2 is UCS2. RestComm can include this header for outgoing SIP MESSAGE and SMSC will include it in actual SMS. For incoming SMS, the SIP MESSAGE from SMSC will include this optional header.

X-SMS-VALIDITY

This parameter holds the SMS validity period in case of re-try. Validity Period is in the format yyyyy-mm-dd hh:mm:ss. RestComm can include this header for outgoing SIP MESSAGE and SMSC will respect the validity period for failed delivery (re-try). For incoming SMS, the SIP MESSAGE from SMSC will include this optional header.

X-M-CLASS

This parameter holds the message class bits of DCS; 0 (directly to display, flash), 1 (to mobile), 2 (to SIM) or 3 (to SIM toolkit). This is not yet supported in the current version of SMSC.

X-REG-DELIVERY

This is registered_delivery parameter and is used to request a SMSC delivery receipt and/or SME originated acknowledgements. This is set to 1 to get delivery receipt for success or failure.

X-SMSC-ID

This parameter holds the smsc-id of the connection that received the message. If the incoming SMS was from ESME, smsc-id will be the name of the ESME. In case of messages coming in from SIP or SS7, this value will be null.

X-DELIVERY_TIME

This is time the message was received by SMSC. Format will be "yyyyy-mm-dd hh:mm:ss".

Chapter 10. SMPP Simulator Tool

SMPP Simulator is a GUI application for testing of the SMPP connection to SMSC GW and understanding its functionality. The test cases will allow you to test SMSC GW with different scenarios. To know more about Running the SMPP Simulator please refer to Running the SMPP Simulator in this guide.

10.1. SMPP Simulator Tool general parameters

This chapter will describe the first step of SMPP Simulator configuring - establishing a connection to SMSC GW.

Default SMPP Simulator settings fit to SMSC GW settings when SMSC GW is started in "simulator" mode. SMSC GW is configured in this case for a single server ESME with SystemId "test", password "test", SMPP listening port 2776, bind type "transceiver" and ESME address_range=""6666" (this means that bind_request must contain address_range "6666", all SMPP Simulator originated messages must have source address "6666" and all messages for destination "6666" address will be routed to to this ESME). SMPP Simulator must play the "client" role in this case.

If SMPP Simulator is run at the same host as SMSC GW, you can just connect as it is. If not you need to press "Configure" button firstly and then update "SMSC host" parameter with a proper SMSC GW IP address value and press "OK" button. Then to connect to SMSC GW press "Run test" button and then in the new opened window "Start a session" button. A message "Session has been successfully started" must occur in the upper half of the window.

If SMSC GW has other settings you need before of connecting configure SMPP Simulator. Press "Configure" button to see "SMPP general parameters" form. Then configure in a new opened form "SystemId", "Password", "SMSC host", "SMSC port" and "SMPP bind type" (TRANSCIEVER, TRANSMITTER or RECIEVER) parameters.

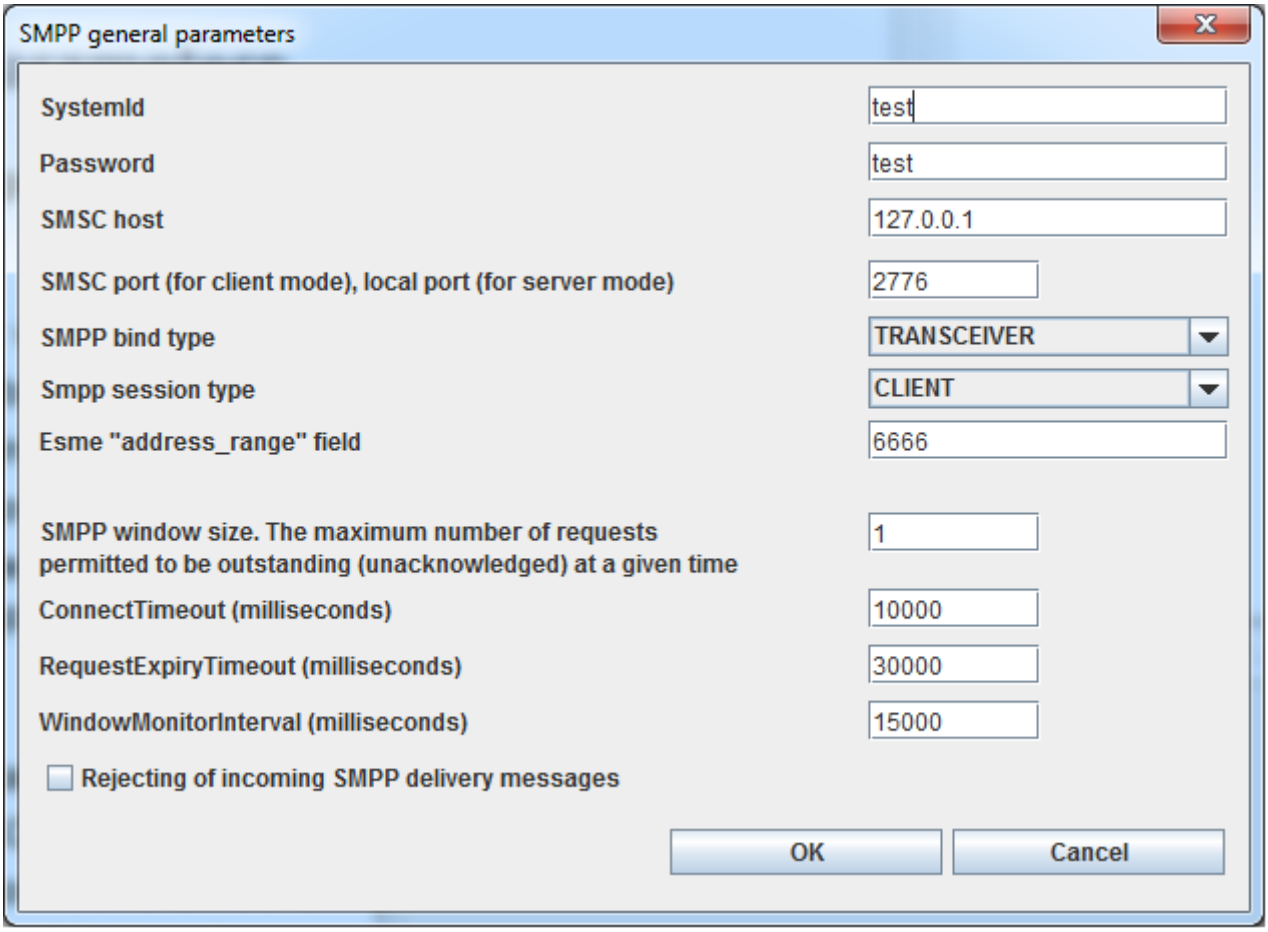


Figure 7. SMPP Simulator "SMPP general parameters" form

SMPP session type is usually "CLIENT". This means that SMSC GW has ESME "SERVER" type and SMPP will create TCP connection to SMSC GW ("SMSC port" is used as a SMSC GW side port). SMSC GW must be configured to allow any client port (ESME "Port" parameter must be set to "-1" value when ESME creating). When SMSC GW has ESME configured as "CLIENT", then you need to configure SMPP Simulator "SMPP Session type" as "SERVER". In this case SMPP Simulator will listen "SMSC port" port for incoming connections and SMSC GW will try to establish a TCP connection to SMPP Simulator. "Esme "address_range" field" means "address_range" field value that will be put into bind_submit or bind_data request to establish a connection to SMSC GW.

For fine tuning you can configure SMPP values "SMPP windows size", "ConnectionTimeout", "RequestExpiryTimeout" and "WindowMonitorInterval" parameters. But usually it is not needed.

If you configure "Rejecting of incoming SMPP delivery messages" parameter to "true" than SMPP Simulator will reject all incoming from SMSC GW deliver_sm messages (with cause 1 - ESME_RINVMSGLEN).

After finishing of configuring press "OK" button and then to connect to SMSC GW press "Run test" button and then in the new opened window "Start a session" button. A message "Session has been successfully started" must occur in the upper half of the window.

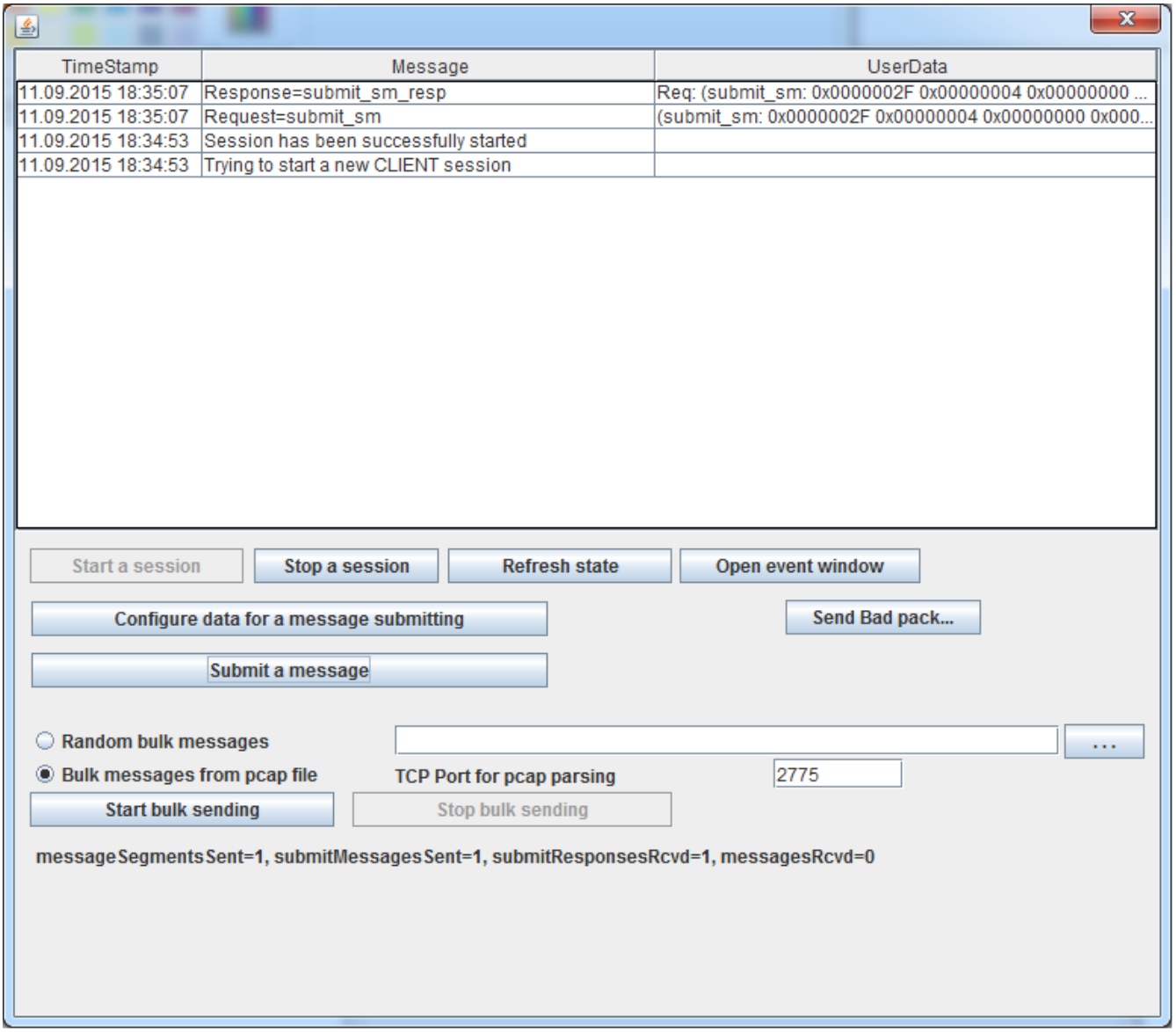


Figure 8. SMPP Simulator main testing form

SMPP Simulator does not store configured values and after SMPP Simulator restart you need to configure all values again.

10.2. SMPP Simulator Tool message parameters

We will describe in this chapter how to receive and send a single message. Bulk message task (sending of very many messages) is described in the next chapter.

I assume that SMPP Simulator is already connected to SMSC GW.

The information of all received message will be put in the top half of the application form. SMPP Simulator will accept all received messages if the option "Rejecting of incoming SMPP delivery messages" is not set to true.

There is a button "Open event window" in the form which opens a new window with a selected record (in the top half of the form) full data. This is useful to check the content and fields of incoming messages and responses from SMSC GW.

Before message sending press to the button "Configure data for message submitting" to open "SMPP message parameters form".

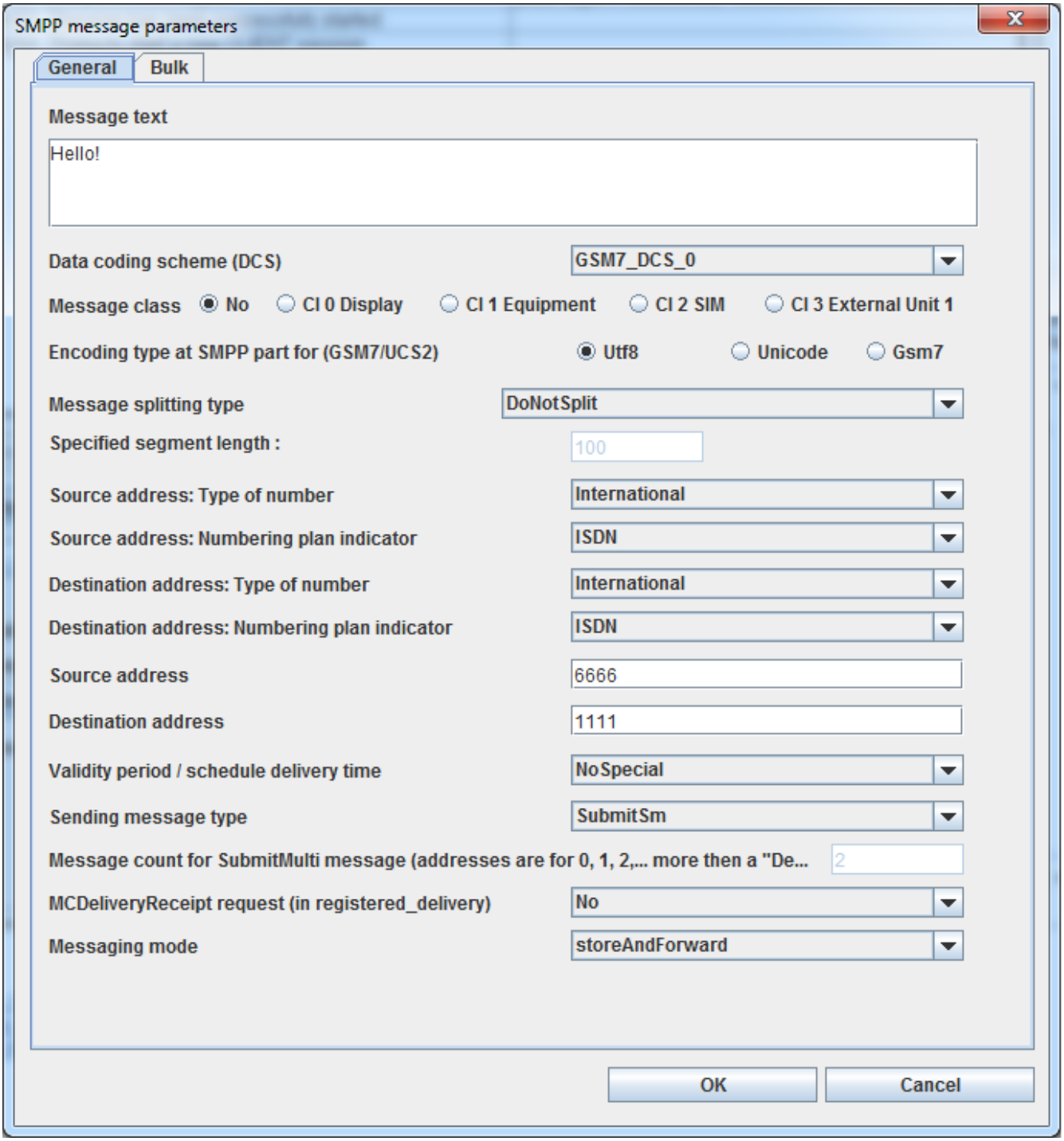


Figure 9. SMPP Simulator "SMPP message parameters" form

The first point of which you must take a care is - "Source address" and "Destination address". For SMSC GW Simulator profile for SMPP Simulator address "6666" is assigned , for SIP - "5555" and all other addresses for SS7 network. This means that when you want to send a message to a SS7 network then you need to configure "Source address" to "6666" and "Destination address" to value other then "5555" and "6666" (for example "1111"). And when you want to send a message to a SIP connector then you need to configure "Source address" to "6666" and "Destination address" to "5555". For other cases "Source address" and "Destination address" will be others. For "Type of number" and "Numbering plan indicator" parameters you can leave default values ("International" and "ISDN") for most cases. Change them for the usage of specific numbering.

And then you can put a desired message text, press "OK" and then "Submit a message" buttons. A message should be submitted to SMSC GW for delivering.

Other parameters in "General" TAB of "SMPP message parameters form" allow you to specify sending message parameters. Here is a list of configurable parameters :

1. Data coding scheme (DCS): you can configure GSM7_DCS_0 (GSM7 default encoding - that covers latin characters, digits and also some other little subset of characters), GSM8_DCS_4 (binary messages, will be encoded by ISO-8859-1 charset in SMPP part) and UCS2_DCS_8 (UCS2 messages - that covers all characters). This parameter affects to "data_coding" parameter in submit_sm message (and further encoding when delivering to SS7 network) but does not affect to message text encoding at SMPP part (For this there is another option).
2. Message class: means a GSM message class that is encoded in DCS field in SS7 messages, encoded as Destination Address Subunit TLV optional parameter.
3. Encoding type at SMPP part for (GSM7/UCS2) (values: Utf8 / Unicode / Gsm7): Encoding / decoding that will be applied to message text for DCS=0 (GSM7) and 8 (UCS2). For DCS=4 (GSM8) encoding is always ISO-8859-1 because of this DCS is meant as binary.
4. Message splitting type: this option is used when we are sending a long message which must be splitted into several segments. A message can we sent to SMSC GW as a single message (SMSM GW will split it itself when delivering the message to SS7 destination). Use in this case "DoNotSplit" option. A message can be splitted by SMPP simulator and sent to SMSC GW by a set of several messages. You can split messages by using of two possible encoding styles: "SplitWithParameters" (by using of optional Tlv parameters SAR_MSG_REF_NUM, SAR_TOTAL_SEGMENTS and SAR_SEGMENT_SEQNUM) and "SplitWithUdh" (by using of "Concatenated short messages" User data header (UDH)). Another option is a length of each message segment - DefaultSegmentLength (each segment will have a length that is long enough to be sent to SS7 subscriber, the exact length depends on DCS) and SpecifiedSegmentLength (you have to specify a desired length in the field "Specified segment length").
5. Validity period / Schedule delivery time: Default value is "No special" that means that neither "Validity period" nor "Schedule delivery time" will be added to a sending messages. If you configure a value "ValidityPeriod_5min" then a message will be sent with "Validity period" 5 minutes from the sending time. This means that SMSC GW will make delivery attempts only within 5 minutes from a message has come to SMSC GW. If you configure a value "ValidityPeriod_2hours" then a message will be sent with "Validity period" 2 hours from the sending time. This means that SMSC GW will make delivery attempts only within 2 hours from a message has come to SMSC GW. If you configure a value "ScheduleDeliveryTime_5min" then a message will be sent with "Schedule Delivery Time" 5 minutes from the sending time. This means that SMSC GW will make the first delivery attempt only in 5 minutes from a time when a message has come to SMSC GW.
6. Message sending time: outgoing messages will be sent by submit_sm, data_sm, deliver_sm or submit_multi SMPP messages. submit_sm, data_sm, and submit_multi can be used when SMPP Simulator plays an SMPP CLIENT role (the most common use). deliver_sm can be used when SMPP Simulator plays an SMPP SERVER role. When usage of submit_multi message the message will be sent to several destination addresses. The exact count of destination addresses is set by "Message count for for Submit_milti message" parameter (default value is 2). Destination addresses in submit_multi will be: next address is by 1 more then a previous one. For example if

"Destination address" has value "1111", then submit_multi destination addresses will be "1111", "1112", "1113", ...

7. MCDeliveryReceipt request: set the SMPP field "registered_delivery" about should SMSC GW send back delivery receipts back to SMPP simulator or not. Default value is "No" - no delivery receipts will be generated. Other possible values: "onSuccessOrFailure" (delivery receipts will be generated in any case), "onFailure" (delivery receipts will be generated only when delivery failure) and "onSuccess (delivery receipts will be generated only when delivery success)"
8. Messaging mode: set the SMPP esm_class field with value "Messaging Mode". Possible values are: "storeAndForward" (default - all messages are stored into databases and several delivery attempts after failure), "datagramm" (message are not stored in a database, only one message delivery retry), "transaction" (also only one retry and no database storage but SMSC GW will return in response the result of message delivery (success / failure)), "defaultSmscMode" (one of mentioned above modes which is SMSC GW default).

"Submit message" button initiate of sending of one message to SMSC GW. "Send bad packet" button sends to SMSC GW a malformed packet (for testing purposes).

10.3. SMPP Simulator Tool - sending of bulk messages

SMPP Simulator can send bulk messages in two modes:

1. A random bulk messages set
2. Bulk messages that are taken from some PCAP trace file

10.3.1. Sending of random bulk messages

Before a sending start you need to configure parameters like it is described in SMPP Simulator Tool message parameters except of a destination address.

Then you need to configure extra options in the TAB "Bulk" of "SMPP Message parameters form" (to see this form press "Configure data for a message submitting" button).

"Destination address range start" and "Destination address range end" parameters describe destination addresses that will be (randomly) assigned to outgoing messages. If you configure start and end field with the same value all messages will be sent to the same address. If you configure a big range this means that most messages will be addressed to different addresses.

"Bulk messages per a second" parameter sets the rate with which messages will be sent to SMSC GW.

After setting of parameters you need to select "Random bulk messages" option and press "Start bulk sending" button. To stop bulk message sending press "Stop bulk sending" button. When a bulk message sending process no records will not be put into top half of the form (because there are too many messages when bulk sending).

When a random bulk message sending process, many messages will contain the message text that is configured. But some messages will be supplied by a long length message text. This is done for simulating of long message splitting at SMSC GW side.

10.3.2. Sending of bulk messages from a PCAP file

SMPP Simulator can use for bulk messages data a source of captured SMPP traffic in a PCAP Wireshark trace data file. The target of this solution is a possibility to simulate some desired traffic to SMSC GW.

In order to achieve it you need to select the option "Bulk messages from pcap file", select a file name by pressing of ". . ." button, and specify "TCP port for TCAP parsing". All TCP messages which have a destination TCP port as here configured, will be seached in PCAP file, then be parsed and parsed messages will be sent to SMSC GW.

After setting of parameters you need to press "Start bulk sending" button. To stop bulk message sending press "Stop bulk sending" button.

Appendix A: Error Codes

The following error codes are used to indicate the Error Code value for the last delivery attempt in table field `SM_STATUS`. Some error codes are reserved for future use and will be put to use in future versions of RestComm SMSC Gateway.

```
    SUCCESS(0),

    UNKNOWN_SUBSCRIBER(1),
    UNDEFINED_SUBSCRIBER(2),
    ILLEGAL_SUBSCRIBER(3),
    TELESERVICE_NOT_PROVISIONED(4),
    CALL_BARRED(5),
    CUG_REJECTED(6),
    FACILITY_NOT_SUPPORTED(7),
    ABSENT_SUBSCRIBER(8),
    ABSENT_SUBSCRIBER_MWDSET(9),
    SENDING_SM_FAILED(10),
    MESSAGE_QUEUE_FULL(11),
    SYSTEM_FAILURE(12),
    DATA_MISSING(13),
    UNEXPECTED_DATA(14),

    ERROR_IN_MS(15),
    MS_NOT_EQUIPPED(16),
    MEMORY_FULL(17),
    SC_CONGESTION(18),
    MS_NOT_REG_IN_SC(19),
    INVALID_SME_ADDRESS(20),
    UNKNOWN_SMC(21),
    ILLEGAL_EQUIPMENT(22),
    USER_BUSY(23),
    //Power off (24вЂ“47: CDMA network reserved). 24-27
    //Reserved.   25-29
    //
    HLR_MESSAGE_DECODING_ERROR(30),
    MSC_MESSAGE_DECODIG_ERROR(31),
    INFORM_SC_MESSAGE_DECODING_ERROR(32),
    //'Not obtain enough routing information' ??? what?
    NOT_ENOUGH_ROUTING_INFORMATION(33),
    UNEXPECTED_DATA_FROM_HLR(34),
    UNEXPECTED_DATA_FROM_MSC(35),
    UNKNOWN_ERROR_FROM_MSC(36),
    UNKNOWN_ERROR_FROM_HLR(37),
    //Reserved(38-39),
    BAD_TYPE_OF_NUMBER(38),
    RESERVED_39(39),
    /**
     * HLR does not send acknowledgment after routing request is sent(40),
```

```
     */
    HLR_REJECT_AFTER_ROUTING_INFO(40),
    /**
    * HLR does not send acknowledgment after the message of setting flag bit is
sent(41),
    */
    HLR_REJECT_AFTER_FLAG_BIT(41),
    //Reserved(42-44),
    RESERVED_42(42),
    RESERVED_43(43),
    RESERVED_44(44),

    MAP_SERVER_VERSION_ERROR(45),
    HLR_VERSION_NEGOTIATION_ERROR(46),
    RESERVED_47(47),
    /**
     * MSC does not send acknowledgment(48),
     */
    MSC_NO_ACK(48),
    /**
     * HLR does not send acknowledgment(49),
     */
    HLR_NO_ACK(49),
    /**
     * GIW does not send acknowledgment(50),
     */
    GIW_NO_ACK(50),
    MSC_REFUSES_SM(51),
    HLR_REFUSES_SM(52),
    GIW_REFUSES_SM(53),
    //No response from the SGSN,
    NO_RESPONSE_FROM_SGSN(54),

    SGSN_REGUSES_SM(55),
    HLR_SYSTEM_ERROR(56),
    MSC_SYSTEM_ERROR(57),
    SGSN_SYSTEM_ERROR(58),
    RESERVED_59(59),
    RESERVED_60(60),
    /**
     * Delivery error due to MAP Server flow control(61),
     */
    MAP_SERVER_FLOW_CONTROLL_ERROR(61),
    /**
     * Delivery error due to MTI Server flow control(62),
     */
    MTI_SERVER_FLOW_CONTROLL_ERROR(62),
    /**
     * SCCP of DSP or STP unable to send the message(63),
     */
    SCCP_UNABLE_TO_SEND(63),
```

```
    INTERFACE_NO_DELIVERY_AUTHORITY(64),
    /**
     * GW does not send acknowledgment(65),
     *
     */
    GW_NO_ACK(65),
    /**
     * Temporary interface error (deregistered or not log in)(66),
     */
    TEMPORARY_INTERFACE_ERROR(66),
    INVALID_INTERFACE(67),
    /**
     * Service interface does not send acknowledgment(68),
     */
    SERVICE_INTERFACE_NO_ACK(68),
    RESERVED_69(69),
    /**
     * Exceeding maximum of messages in L2CacheDaemon specified by license(70),
     */
    EXCEEDED_L2CacheDaemon_MAX_MESSAGE(70),
    /**
     * Deletion of SMs to be imported to the database because of disconnection between
the SMSC and L2CacheDaemon(71),
     */
    DELETION_OF_SM_ON_SMSC_L2CacheDaemon_DISCONNECT(71),
    /**
     * Exceeding the maximum L2CacheDaemon capacity specified by license(72),
     */
    EXCEEDED_L2CacheDaemon_CAPACITY(72),
    /**
     * Some messages in L2CacheDaemon are the same as the messages in memory(73),
     */
    L2CacheDaemon_MESSAGE_DUPLICATION(73),
    L2CacheDaemon_DB_UNAVAILABLE(74),
    L2CacheDaemon_CONGESTED(75),
    /**
     * Error when exporting SMs from memory to L2CacheDaemon(76),
     */
    ERROR_ON_EXPORTING_SM_FROM_MEMORY_TO_L2CacheDaemon(76),
    /**
     * POOL may be full in message delivery(77),
     */
    POOL_FULL_IN_DELIVERY(77),
    /**
     * MT speed exceeds the License threshold by 120%(78),
     */
    MT_SPEED_EXCEEDED(78),
    /**
     * Number of entities that exceed the maximum submission number(Delivery of the SM
failed and the SM is deleted(79),
     */
```

```
    MAX_SM_DELIVERY_RETRY_EXCEEDED(79),
    //Reserved(80 - 127),
    RESERVED_80(80),
    RESERVED_81(81),
    RESERVED_82(82),
    RESERVED_83(83),
    RESERVED_84(84),
    RESERVED_85(85),
    RESERVED_86(86),
    RESERVED_87(87),
    RESERVED_88(88),
    RESERVED_89(89),
    RESERVED_90(90),
    RESERVED_91(91),
    RESERVED_92(92),
    RESERVED_93(93),
    RESERVED_94(94),
    RESERVED_95(95),
    RESERVED_96(96),
    RESERVED_97(97),
    RESERVED_98(98),
    RESERVED_99(99),
    RESERVED_100(100),
    RESERVED_101(101),
    RESERVED_102(102),
    RESERVED_103(103),
    RESERVED_104(104),
    RESERVED_105(105),
    RESERVED_106(106),
    RESERVED_107(107),
    RESERVED_108(108),
    RESERVED_109(109),
    RESERVED_110(110),
    RESERVED_111(111),
    RESERVED_112(112),
    RESERVED_113(113),
    RESERVED_114(114),
    RESERVED_115(115),
    RESERVED_116(116),
    RESERVED_117(117),
    RESERVED_118(118),
    RESERVED_119(119),
    RESERVED_120(120),
    RESERVED_121(121),
    RESERVED_122(122),
    RESERVED_123(123),
    RESERVED_124(124),
    RESERVED_125(125),
    RESERVED_126(126),
    RESERVED_127(127),
```

```
    /**
     * Teleservice facility interaction not supported(128),
     */
    TELESERVICE_FACILITY_INTERACTION_NOT_SUPPORTED(128),
    SM_TYPE_0_NOT_SUPPORTED(129),
    CANNOT_REPLACE_SM(128),
    //Reserved(131вЂ“142),
    RESERVED_131(131),
    RESERVED_132(132),
    RESERVED_133(133),
    RESERVED_134(134),
    RESERVED_135(135),
    RESERVED_136(136),
    RESERVED_137(137),
    RESERVED_138(138),
    RESERVED_139(139),
    RESERVED_140(140),
    RESERVED_141(141),
    RESERVED_142(142),
    UNSPECIFIED_TP_PID_ERROR(143),
    DCS_NOT_SUPPORTED(144),
    SM_TYPE_NOT_SUPPORTED(145),
    //Reserved(146вЂ“158),
    RESERVED_146(146),
    RESERVED_147(147),
    RESERVED_148(148),
    RESERVED_149(149),
    RESERVED_150(150),
    RESERVED_151(151),
    RESERVED_152(152),
    RESERVED_153(153),
    RESERVED_154(154),
    RESERVED_155(155),
    RESERVED_156(156),
    RESERVED_157(157),
    RESERVED_158(158),
    UNSPECIFIED_TP_DCS_ERROR(159),
    OPERATION_NOT_EXECUTED(160),
    //Reserved(161вЂ“174),
    RESERVED_161(161),
    RESERVED_162(162),
    RESERVED_163(163),
    RESERVED_164(164),
    RESERVED_165(165),
    RESERVED_166(166),
    RESERVED_167(167),
    RESERVED_168(168),
    RESERVED_169(169),
    RESERVED_170(170),
    RESERVED_171(171),
```

```
    RESERVED_172(172),
    RESERVED_173(173),
    RESERVED_174(174),
    TPDU_NOT_SUPPORTED(176),
    //Reserved(177вЂ“191),
    RESERVED_177(177),
    RESERVED_178(178),
    RESERVED_179(179),
    RESERVED_180(180),
    RESERVED_181(181),
    RESERVED_182(182),
    RESERVED_183(183),
    RESERVED_184(184),
    RESERVED_185(185),
    RESERVED_186(186),
    RESERVED_187(187),
    RESERVED_188(188),
    RESERVED_189(189),
    RESERVED_190(190),
    RESERVED_191(191),

    SC_BUSY(192),
    NO_SC_SPECIFIED(193),
    SC_SYSTEM_ERROR(194),
    INVALID_SME_ADDRESS_2(195),
    DESTINATION_SME_PROHIBITED(196),
    //Reserved(197-207),
    RESERVED_197(197),
    RESERVED_198(198),
    RESERVED_199(199),
    RESERVED_200(200),
    RESERVED_201(201),
    RESERVED_202(202),
    RESERVED_203(203),
    RESERVED_204(204),
    RESERVED_205(205),
    RESERVED_206(206),
    RESERVED_207(207),
    SIM_SMS_STORAGE_IS_FULL(208),
    /**
     * No SMS storage capability in SIM(209),
     */
    SIM_HAS_NO_SMS_STORAGE(209),
    ERROR_IN_MS_2(210),
    ESME_MEMORY_OVERFLOW(211),
    /**
     * Reserved(212-223),
     *
     */
    RESERVED_212(212),
    RESERVED_213(213),
```

```
    RESERVED_214(214),
    RESERVED_215(215),
    RESERVED_216(216),
    RESERVED_217(217),
    RESERVED_218(218),
    RESERVED_219(219),
    RESERVED_220(220),
    RESERVED_221(221),
    RESERVED_222(222),
    RESERVED_223(223),
    /**
     * Values specific to an application(224-254),
     */
    OCS_ACCESS_NOT_GRANTED(224),
    MPROC_ACCESS_NOT_GRANTED(225),
    MPROC_SRI_REQUEST_DROP(226),
    APP_SPECIFIC_227(227),
    APP_SPECIFIC_228(228),
    APP_SPECIFIC_229(229),
    APP_SPECIFIC_230(230),
    APP_SPECIFIC_231(231),
    APP_SPECIFIC_232(232),
    APP_SPECIFIC_233(233),
    APP_SPECIFIC_234(234),
    APP_SPECIFIC_235(235),
    APP_SPECIFIC_236(236),
    APP_SPECIFIC_237(237),
    APP_SPECIFIC_238(238),
    APP_SPECIFIC_239(239),
    APP_SPECIFIC_240(240),
    APP_SPECIFIC_241(241),
    APP_SPECIFIC_242(242),
    APP_SPECIFIC_243(243),
    APP_SPECIFIC_244(244),
    APP_SPECIFIC_245(245),
    APP_SPECIFIC_246(246),
    APP_SPECIFIC_247(247),
    APP_SPECIFIC_248(248),
    APP_SPECIFIC_249(249),
    APP_SPECIFIC_250(250),
    APP_SPECIFIC_251(251),
    APP_SPECIFIC_252(252),
    APP_SPECIFIC_253(253),
    APP_SPECIFIC_254(254),
    UNSPECIFIED_ERROR_CAUSE(255);
```

Appendix B: Revision History