

Running

Table of Contents

1. Running the Gateway	1
2. Running the Gateway - Simulator Profile	4
2.1. Running USSD Examples in Simulator	4
3. Running the Shell.....	11
4. Running the Graphical User Interface	12
4.1. Connect to a new Instance	12
4.2. Authentication	13

Chapter 1. Running the Gateway

Procedure: Run Restcomm USSD GATEWAY

1. Pre-requisite:

- You must have Restcomm USSD GATEWAY installed as explained in the Installation Guide.
- If you are using the SS7 board on server, you must ensure that the `java.library.path` variable is set to point to the directory containing the native component. Alternatively you can copy it to the JBoss native library path manually.

2. All you have to do to start the Gateway is start the JBoss Application Server. To start the JBoss Server you must execute the `run.sh` (Unix) or `run.bat` (Microsoft Windows) startup script in the installation directory `mobicents-ussdgateway-jboss-5.1.0.GA/bin`. Note that this will start the server in the default profile. The "default" profile is a clean profile where you start from scratch and configure the entire SS7 Stack and USSD Gateway to suit your requirements.
3. Result: If the service started properly you should see the following last few output lines in the Unix terminal or Command Prompt depending on your environment:

```

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

```

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

Procedure: Stop the Gateway

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

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

Chapter 2. Running the Gateway - Simulator Profile

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

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

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

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

2.1. Running USSD Examples in Simulator

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

Procedure: Running Mobicents jSS7 Simulator - USSD Pull Example

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

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

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

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

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

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

This will launch the Simulator GUI Application.

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

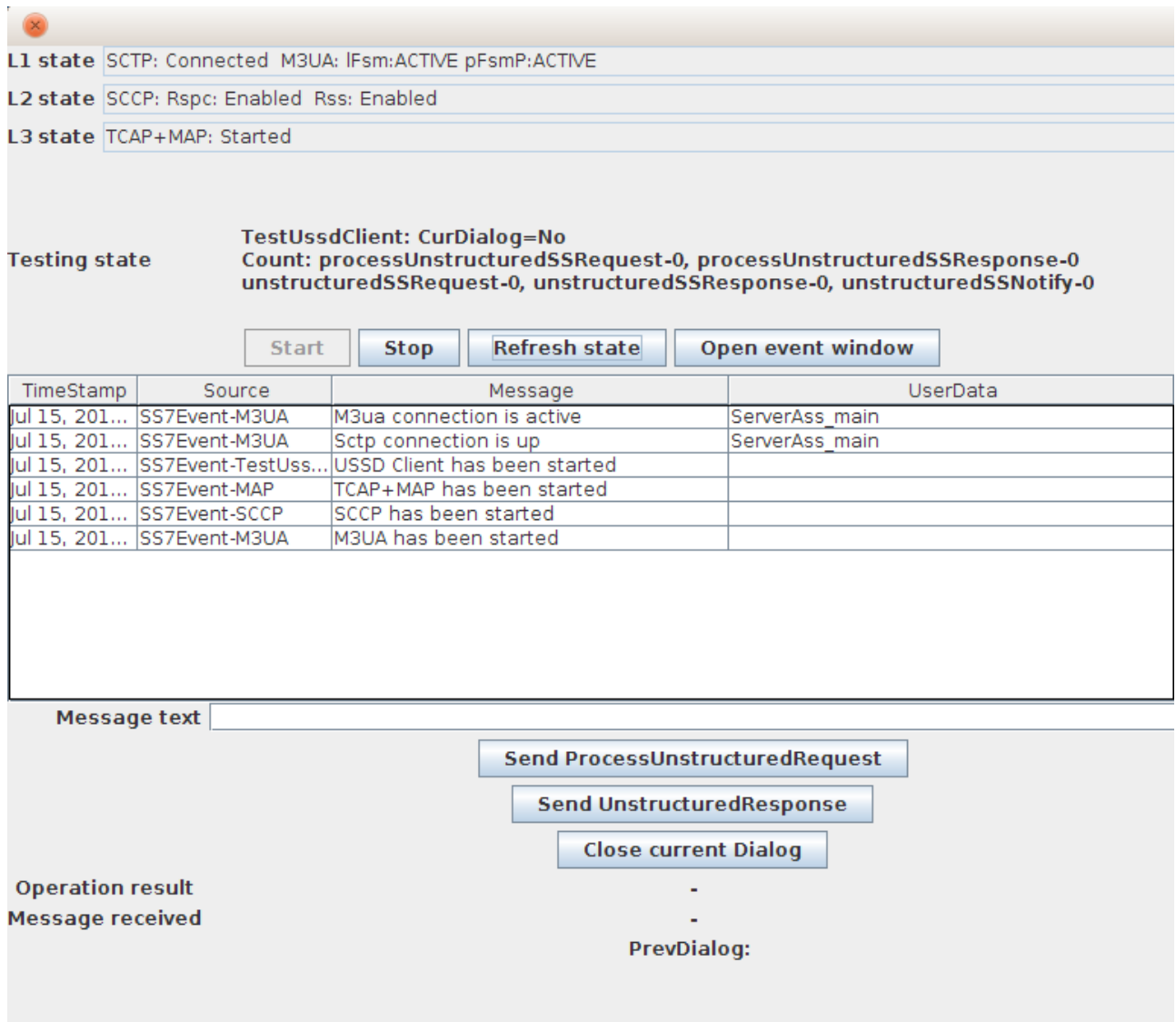


Figure 1. USSD SS7 Simulator - Active

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

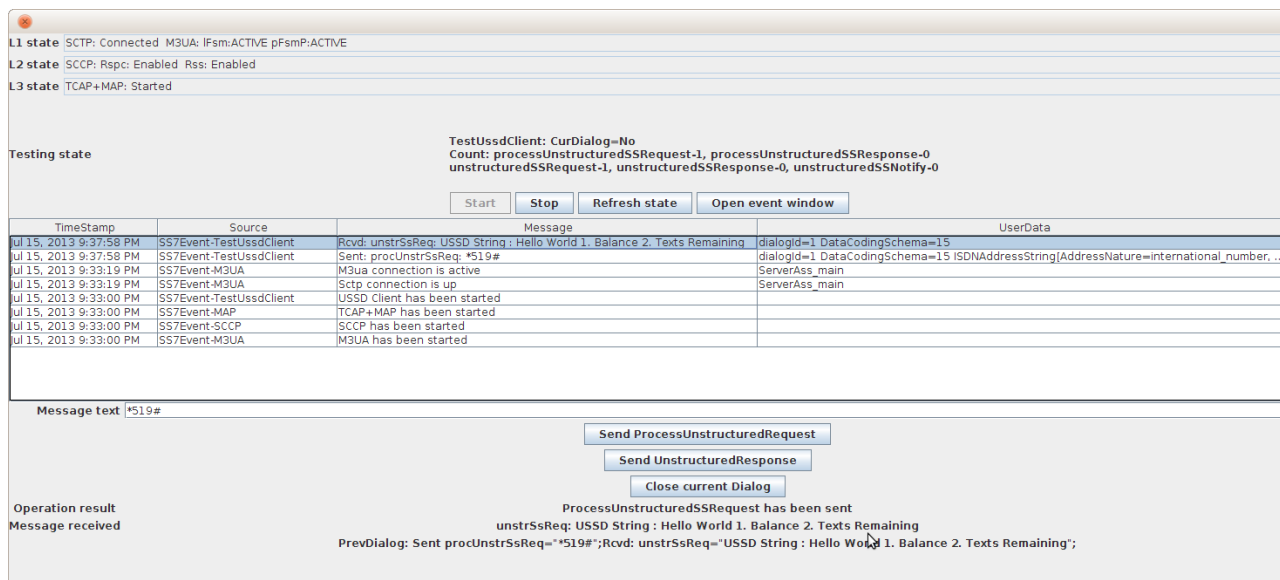


Figure 2. USSD SS7 Simulator - Process Unstructured Request

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

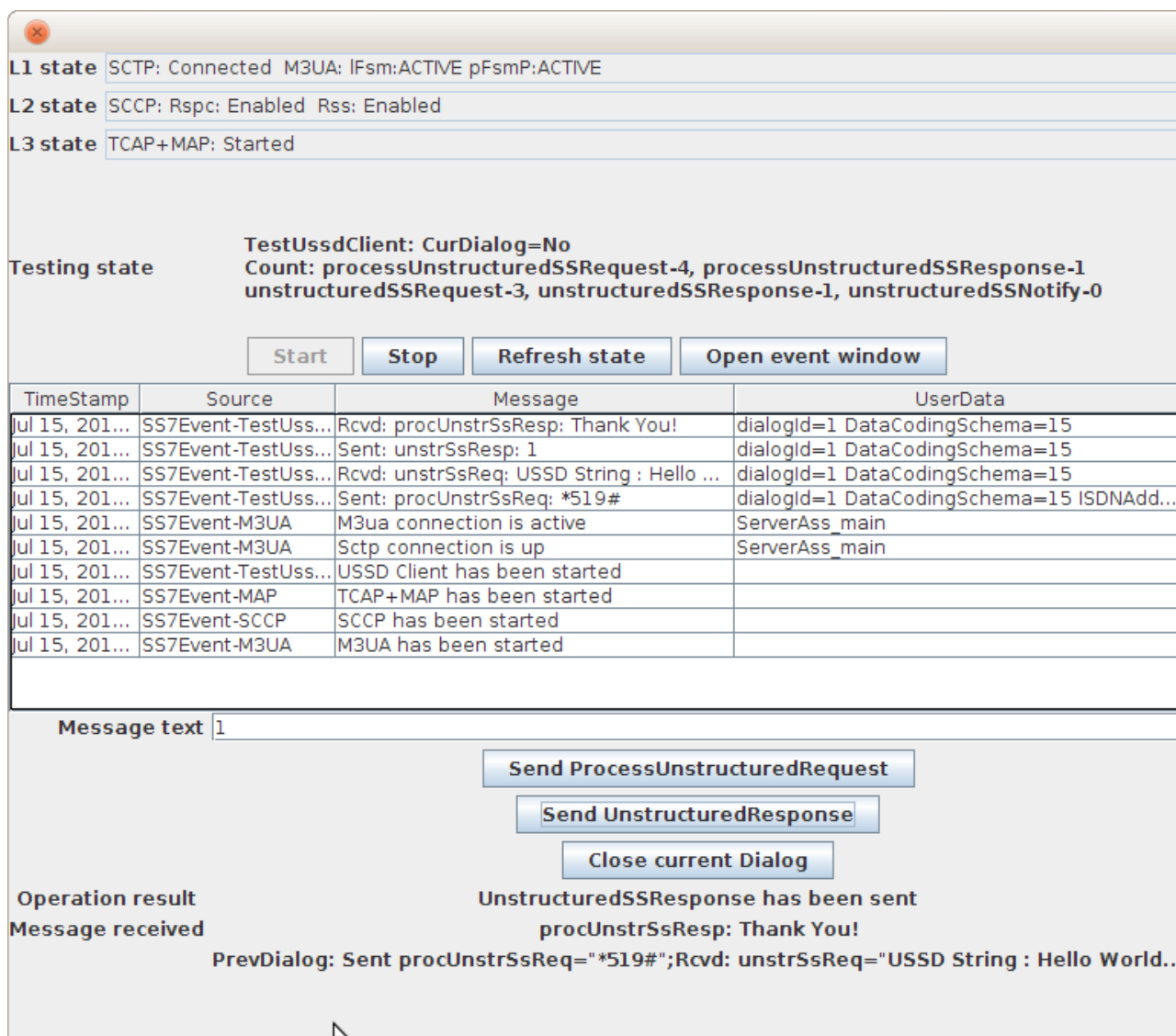


Figure 3. USSD SS7 Simulator - Unstructured Request

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

1. You must first start the Restcomm USSD GATEWAY in simulator profile.

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

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

The screenshot displays the Mobiscents USSD Simulator interface. At the top, it shows the current state: .1 state Sctp: Connected M3UA: IFsm:ACTIVE pFsmP:ACTIVE, .2 state SCCP: Rspc: Enabled Rss: Enabled, and .3 state TCAP+MAP: Started. Below this, the 'Testing state' section shows 'TestUssdServer: CurDialog=1, Pending dialogs=0' and a count of various requests and responses. A row of buttons includes 'Start', 'Stop', 'Refresh state', and 'Open event window'. A table below lists events with columns for TimeStamp, Source, Message, and UserData. The 'Message text' field contains 'send Notify'. A sequence of buttons for sending responses, requests, and notifications is shown, followed by a confirmation message 'UnstructuredSSNotify has been sent' and the 'CurDialog:' label.

TimeStamp	Source	Message	UserData
Feb 10, 2016 ...	SS7Event-TestUssdS...	Sent: unstrSsNotify: send NOTIFY	dialogId=1 DataCodingSchema=15 ISDNAddress...
Feb 10, 2016 ...	SS7Event-M3UA	M3ua connection is active	ServerAss_main
Feb 10, 2016 ...	SS7Event-M3UA	Sctp connection is up	ServerAss_main
Feb 10, 2016 ...	SS7Event-TestUssdS...	USSD Server has been started	
Feb 10, 2016 ...	SS7Event-MAP	TCAP+MAP has been started	
Feb 10, 2016 ...	SS7Event-SCCP	SCCP has been started	
Feb 10, 2016 ...	SS7Event-M3UA	M3UA has been started	

Figure 4. Mobiscents USSD Simulator - Notify

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

- Fill the ISDN field with a preferred ISDN number, for example "1111" is good for SS7 Simulator. Now press "Apply changes".
- Perform "reset" operation. Perform "sendNotify" operation with parameters: String=<Text of your notification>, boolean=false, int=60000 and String=<any random string>. Parameters definition is as below

- 1st String is USSD message that you want to push to mobile
- 2nd Boolean if set to true means USSD Gw will send empty TCAP Begin and try to establish dialog before sending actual message.
- 3rd Int is custom invoke timeout. User must respond within this period else USSD Gw will terminate Dialog and Application will get appropriate error message
- 4th String is random string that is stored at USSD Gw side as custom object.
- When ever response comes back, USSD Gw will include this custom string in XML Payload.
- Perform "close" operation. You will now find a notification at the SS7 Simulator.

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

```
/**
 * Simple MBean interface. This MBean is front end of simple example for ussd
 * push via HTTP.
 */
public interface HTTPPushMBean {

    /**
     * The URI where HTTP Post request is to be submitted. This should point the
     * USSD Gateway. Basically http://USSD-IP:8080/mobicents
     *
     * @param uri
     */
    public void setTargetUri(String uri);

    /**
     * Get the URI pointing to USSD Gateway for push
     *
     * @return
     */
    public String getTargetUri();

    /**
     * Set the MSISDN where USSD Push is to be sent
     *
     * @param isdn
     */
    public void setIsdn(String isdn);

    /**
     * Get the MSISDN where USSD request is to be pushed
     *
     * @return
     */
}
```

```

public String getIsdn();

/**
 * Reset( remove local dialog ) in case something goes wrong
 */
public void reset();

/**
 * Starts dialog if not already started. Sends Unstructured Request. It can
 * be sent multiple times in the same dialog
 *
 * @param ussdRequest
 *         The actual USSD String request
 * @param emptyDialogHandshake
 *         If true, USSD Gateway will first establish Dialog by doing
 *         handshake before sending USSD request. If false the USSD
 *         request will be added in Dialog begin message
 * @param invokeTimeout
 *         Time in milliseconds USSD gateway will wait for user to
 *         respond, if user doesn't respond back within specified time,
 *         USSD Gateway will abort the dialog and send back Abort error
 *         to HTTP App
 * @param userData
 *         User Data to be sent with every request to USSD Gateway which will
be
 *         returned back with response from USSD Gw. This is just in case if
 *         application wants to keep some data at Dialog level, for example
MSISDN
 *
 *
 * @throws Exception
 */
public void sendRequest(String ussdRequest, boolean emptyDialogHandshake, int
invokeTimeout, String userData) throws Exception;

/**
 * Starts dialog if not already started. Sends Notify Request. It can be
 * sent multiple times in the same dialog
 *
 * @param ussdRequest
 *         The actual USSD String request
 * @param emptyDialogHandshake
 *         If true, USSD Gateway will first establish Dialog by doing
 *         handshake before sending USSD request. If false the USSD
 *         request will be added in Dialog begin message
 * @param invokeTimeout
 *         Time in milliseconds USSD gateway will wait for user to
 *         respond, if user doesn't respond back within specified time,
 *         USSD Gateway will abort the dialog and send back Abort error
 *         to HTTP App
 * @param userData

```

```

    *      User Data to be sent with every request to USSD Gateway which will
be
    *      returned back with response from USSD Gw. This is just in case if
    *      application wants to keep some data at Dialog level, for example
MSISDN
    * @throws Exception
    */
    public void sendNotify(String ussdRequest, boolean emptyDialogHandshake, int
invokeTimeout, String userData) throws Exception;

    /**
    * USER Abort the underlying MAP Dialog
    *
    * @throws Exception
    */
    public void abort() throws Exception;

    /**
    * Close the underlying MAP Dialog. This will send TCAP End to peer
    *
    * @throws Exception
    */
    public void close() throws Exception;

    /**
    * Return current status of service - what has been sent, what has been
    * received etc.
    *
    * @return
    */
    public String getStatus();
}

```

Chapter 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 *mobicents-ussdgateway-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 Mobicents SS7 Stack User Guide.

Chapter 4. Running the Graphical User Interface

Open a Web Browser and navigate to <http://localhost:8080/mobicents-management/>. 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 USSD GATEWAY you will find the tabs for JAIN-SLEE, JMX, SS7 and USSD 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 USSD Gateway.

Switch to the USSD 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, Routing Rules, Metrics). 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.

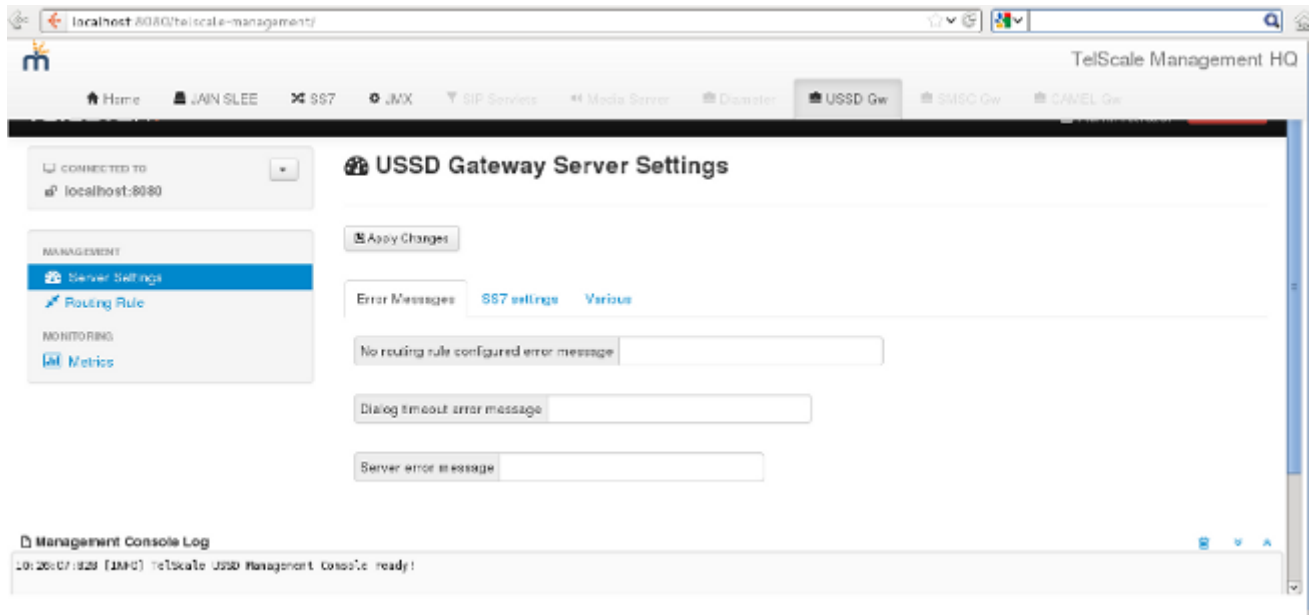


Figure 5. GUI - Restcomm USSD GATEWAY

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.

4.2. Authentication

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

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

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

For detailed instructions and to know more about JBoss Security Framework please refer to the JBoss Installation Guide [here](#).



Default 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 *mobicents-ussdgateway-<version>/jboss-5.1.0.GA/server/<profile>/conf/props/*