

Running

# Table of Contents

Running Restcomm SS7 Service.....	1
Running Restcomm Signaling Gateway.....	3
Start the Gateway .....	3
Stop the Gateway .....	4
Bind the Gateway to IP.....	4
Running jSS7 as a Standalone library .....	4
Running the Shell `Shell` .....	4
Start the Shell Client .....	4
Connect to Managed Instance.....	5
Disconnect .....	5
Authentication and Audit Logs.....	5
Command-line Completion .....	7
Running the Graphical User Interface .....	8
Connect to a new Instance.....	9
Authentication .....	9
Running Restcomm jSS7 Simulator .....	9
Running the Simulator Locally .....	10
Running SS7 Simulator remotely.....	12

You must ensure that you follow the steps outlined in the Restcomm jSS7 Installation Guide to install the Components. As explained in the Installation Guide, you may install jSS7 as a standalone library or as a JBoss AS SS7 Service or as a Signaling Gateway. Once installed, you can run the different Components as described in the sections below.

## Running Restcomm SS7 Service

### *Procedure: Start SS7 Service*

1. Pre-requisite: You must have SS7 Service deployed in the JBoss Application Server as explained in the Installation Guide.
2. All you have to do to start the Service is start the JBoss AS. This will automatically start the SS7 Service. To start the JBoss Server you must execute the *run.sh* (Unix) or *run.bat* (Microsoft Windows) startup script in the *<jboss\_install\_directory>/bin* folder (on Unix or Windows).
3. Result: If the service started properly you should see following lines in the Unix terminal or Command Prompt depending on your environment:

```

14:07:47,580 INFO [TomcatDeployment] deploy, ctxPath=/admin-console
14:07:47,650 INFO [config] Initializing Mojarra (1.2_12-b01-FCS) for context
'/admin-console'
14:07:49,980 INFO [TomcatDeployment] deploy, ctxPath=/
14:07:50,032 INFO [TomcatDeployment] deploy, ctxPath=/jmx-console
14:07:50,235 INFO [ManagementImpl] SCTP configuration file path /home/vinu/jboss-
5.1.0.GA/server/default/data/SCTPManagement_sctp.xml
14:07:50,262 INFO [ManagementImpl] Started SCTP Management=SCTPManagement
WorkerThreads=0 SingleThread=true
14:07:50,269 INFO [SelectorThread] SelectorThread for Management=SCTPManagement
started.
14:07:50,422 INFO [M3UAManagement] M3UA configuration file path /home/vinu/jboss-
5.1.0.GA/server/default/data/Mtp3UserPart_m3ua.xml
14:07:50,512 INFO [M3UAManagement] Started M3UAManagement
14:07:50,605 INFO [SccpStackImpl-SccpStack] Starting ...
14:07:50,665 INFO [Router] SCCP Router configuration file path /home/vinu/jboss-
5.1.0.GA/server/default/data/SccpStack_sccprouter.xml
14:07:50,667 INFO [Router] Started SCCP Router
14:07:50,668 INFO [SccpResource] SCCP Resource configuration file path
/home/vinu/jboss-5.1.0.GA/server/default/data/SccpStack_sccpresource.xml
14:07:50,669 INFO [SccpResource] Started Sccp Resource
14:07:50,671 INFO [SccpStackImpl-SccpStack] Starting routing engine...
14:07:50,671 INFO [SccpStackImpl-SccpStack] Starting management ...
14:07:50,671 INFO [SccpStackImpl-SccpStack] Starting MSU handler...
14:07:50,696 INFO [ShellExecutor] Starting SS7 management shell environment
14:07:50,714 INFO [ShellExecutor] ShellExecutor listening at /127.0.0.1:3435
14:07:50,776 INFO [TCAPStackImpl] Starting
...org.mobicenss7protocols.tcapi.TCAPProviderImpl@14d18a1
14:07:50,776 INFO [TCAPProviderImpl] Starting TCAP Provider
14:07:50,776 INFO [TCAPProviderImpl] Registered SCCP listener with address 8
14:07:51,012 INFO [SS7Service] [||||| Mobicenss7 3.0.0 service started
|||||]
14:07:51,156 INFO [Http11Protocol] Starting Coyote HTTP/1.1 on http-127.0.0.1-8080
14:07:51,177 INFO [AjpProtocol] Starting Coyote AJP/1.3 on ajp-127.0.0.1-8009
14:07:51,184 INFO [ServerImpl] JBoss (Microcontainer) [5.1.0.GA (build:
SVNTag=JBoss_5_1_0_GA date=200905221053)] Started in 40s:691ms

```

4. If you are starting Restcomm-jSS7-3.0.0-SNAPSHOT for the first time, SS7 is not configured. You need to use the Shell Client to connect to Restcomm-jSS7-3.0.0-SNAPSHOT . Using CLI you can configure how service interacts with SS7 network, that is you configure either installed SS7 card and its native library , or **M3UA** layer. You can also use the GUI to achieve the same. Once configured, the state and configuration of SS7 is persisted which stands server re-start.

#### *Procedure: Stop SS7 Service*

1. To stop the SS7 service, 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 `<jboss_install_directory>/bin` directory (on Unix or Windows).
2. If the Service 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
```

# Running Restcomm Signaling Gateway

## Start the Gateway

*Procedure: Start the Restcomm Signaling Gateway on*

1. Change the working directory to the installation directory (the one into which the zip file's contents was extracted to)

```
downloads]$ cd restcomm-jss7-<version>/ss7/restcomm-ss7-sgw
```

2. (Optional) Ensure that the *bin/run.sh* script is executable.

```
restcomm-ss7-sgw$ chmod +x bin/run.sh
```

3. Execute the *run.sh* Bourne shell script.

```
restcomm-ss7-sgw$ ./bin/run.sh
```

4. In the Linux terminal, the Restcomm Signaling Gateway has started successfully if the last line of output is similar to the following

```
15:51:18,247 INFO [MainDeployer] [[[[[[[[ Mobicents Signaling Gateway:  
release.version= Started ]]]]]]]]
```

*Procedure: Start the Restcomm Signaling Gateway on*

1. Using Windows Explorer, navigate to the *restcomm-jss7-<version>/ss7/restcomm-ss7-sgw/bin* subfolder in the installation directory.
2. The preferred way to start the Restcomm Signaling Gateway is from the Command Prompt. The command line interface displays details of the startup process, including any problems encountered during the startup process.

Open the Command Prompt via the Start menu and navigate to the correct folder:

```
C:\Users\<user>\My Downloads>cd "restcomm-jss7-<version>\ss7\restcomm-ss7-sgw"
```

3. Start the Gateway by executing the *run.bat* file:

- *run.bat* batch file:

```
C:\Users\<user>\My Downloads\mms-standalone<version>>bin\run.bat
```

## Stop the Gateway

The only option to stop the gateway is by pressing **Ctrl c** and bringing down the JVM or killing the process.

## Bind the Gateway to IP

Using **run.sh** without any arguments binds the gateway to **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 server to **115.252.103.220** you will use the command as below:

```
restcomm-ss7-sgw$ ./bin/run.sh -b 115.252.103.220
```

## Running jSS7 as a Standalone library

If you do not wish to use the JBoss Container and do not require JSLEE RAs you can configure and run the Stack as a Standalone library. For more details, refer to [\[configuring\\_standalone\]](#)

## Running the Shell`Shell`

### Start the Shell Client

Shell client can be started with following command from *\$JBOSS\_HOME/bin*:

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

Once console starts, it will print following like information:

```
=====
Mobicents SS7 Management Shell Bootstrap Environment
=====
mobicents>
```

The **ss7-cli** script supports the following options

```
Usage: SS7 [OPTIONS]
Valid Options
-v          Display version number and exit
-h          This help screen
```

## Connect to Managed Instance

Shell needs to connect to managed instance. Command to connect has following structure:

```
connect <IP> <PORT>
```

*Example 1. Connect to remote machine*

```
mobicents>connect 10.65.208.215 3435

mobicents(10.65.208.215:3435)>
```



Host IP and port are optional, if not specified, shell will try to connect to **127.0.0.1:3435**

## Disconnect

Command to disconnect has following structure:

```
ss7 disconnect
```

*Example 2. Disconnect*

```
mobicents(10.65.208.215:3435)>ss7 disconnect

Bye
mobicents>
```

## Authentication and Audit Logs

### Security

Security is a fundamental requirement of any Telecom application. You must control access to your

SS7 network and restrict who is allowed to access what and perform what operations.

Restcomm jSS7 CLI Security is based on the JBoss Security Framework. The JBoss Security framework provides support for a role-based declarative security model as well as integration of custom security via a security proxy layer. The default implementation of the declarative security model is based on Java Authentication and Authorization Service (JAAS) login modules and subjects.

#### *Procedure: Enable Security*

1. Add a new parameter named “securityDomain” to the "ShellExecutor" bean in the configuration file `jboss-5.1.0.GA/server/default/deploy/restcomm-ss7-service/META-INF/jboss-beans.xml` and save the changes.

```
<property name="securityDomain">java:/jaas/jmx-console</property>
```

2. Configure the security domain in the file `jboss-5.1.0.GA/server/default/conf/login-config.xml` following the instructions in the JBoss Admin Guide.
3. Create entries for user id and password in the file `[path]_ jboss-5.1.0.GA/server/default/conf/props/jmx-console-users.properties_` for every user allowed to access the CLI.

#### *Procedure: Disable Security*

1. Delete all configurations created as mentioned above and remove the parameter “securityDomain” from the "Shell Executor" bean defined in `jboss-5.1.0.GA/server/default/deploy/restcomm-ss7-service/META-INF/jboss-beans.xml`.

If you would like to read more about the JBoss Security Framework, please refer to the JBoss Admin Guide available in their website.

## **Audit Configuration**

If security is enabled then you can log the operations performed by every user.

#### *Procedure: Enable Audit*

1. Add a new appender to the file `jboss-5.1.0.GA/server/default/conf/jboss-log4j.xml` as below:

```
<appender name="AUDIT" class="org.jboss.logging.appender.DailyRollingFileAppender">
  <errorHandler class="org.jboss.logging.util.OnlyOnceErrorHandler"/>
  <param name="File" value="${jboss.server.log.dir}/audit.log"/>
  <param name="Append" value="true"/>
  <param name="DatePattern" value="'. 'yyyy-MM-dd"/>
  <layout class="org.apache.log4j.PatternLayout">
    <param name="ConversionPattern" value="%d %-5p [%c] (%t:%x) %m%n"/>
  </layout>
</appender>
```

2. Add a new category to the file `jboss-5.1.0.GA/server/default/conf/jboss-log4j.xml` as below:



```
<category name="org.jboss.security.audit.providers.LogAuditProvider"
additivity="false">
  <priority value="TRACE"/>
  <appender-ref ref="AUDIT"/>
</category>
```

When security and audit is enabled, a sample audit log looks like this:

```
2012-11-28 22:17:27,005 TRACE [org.jboss.security.audit.providers.LogAuditProvider]
(pool-19-thread-8:) [Success]message=login success;principal=admin;
2012-11-28 22:17:47,659 TRACE [org.jboss.security.audit.providers.LogAuditProvider]
(pool-19-thread-1:) [Info]response=Successfully added client
Association=SCTPAssoc1;principal=admin;command=sctp association create SCTPAssoc1
CLIENT 127.0.0.1 2775 127.0.0.1 2776;
2012-11-28 22:18:06,773 TRACE [org.jboss.security.audit.providers.LogAuditProvider]
(pool-19-thread-3:) [Success]message=logout success;principal=admin;
```

## Command-line Completion

Restcomm jSS7 Shell supports Command-line completion (Tab Completion) allowing you to type the first (or first few) character(s) of the command and press tab to fill in the rest of the command. As soon as you enter the CLI (by executing the *ss7-cli.sh* script) you can make use of this feature to view all possible commands.

When you first enter CLI and press the "tab" key, CLI will display all operations permitted in that context. Once you connect to a managed instance and while staying connected if you press the "tab" key it will display all other commands allowed.

```
mobicents> [tab key press]
history connect exit
mobicents>connect 10.65.208.215 3435
mobicents(10.65.208.215:3435)> [tab key press]
sctp          linkset      m3ua          sctp          history      disconnect
```

If you enter the first few characters of a command and press "tab", CLI will automatically fill in the rest of the command or display all possible commands if there is more than one command beginning with the characters entered by you.

```
mobicents(10.65.208.215:3435)>sctp [tab key press]
server        association  --help
mobicents(10.65.208.215:3435)>sctp
```

In addition, help files are also available for every command using the *--help* option. The help files provide details of the command including possible parameters and examples of usage if applicable.

```
mobicents(10.65.208.215:3435)>sctp --help
```

Name

sctp

Manage M3UA - SCTP

SYNOPSIS

sctp server [create | destroy | start | stop | show] [parameters]

sctp association [create | destroy | show] [parameters]

parameters

Command Line parameters.

DESCRIPTION

This command is used to manage M3UA - SCTP. You can create, destroy, start and stop a SCTP Server and view the Server configuration using the sctp server command. You can create, destroy and view SCTP Associations using the sctp association command.

SEE ALSO

sctp server create, sctp server destroy, sctp server start, sctp server stop, sctp server show, sctp association create, sctp association destroy, sctp association show

```
mobicents(10.65.208.215:3435)>
```

## Running the Graphical User Interface

Open a Web Browser and navigate to <http://localhost:8080/jss7-management-console/>. The window will look similar to the figure below. The GUI is divided into three sections:

- A left panel listing the management units (Services, SCTP, M3UA, Linkset, SCCP, TCAP, Alarms, Manage Campaigns and 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. At the top of this panel you will find a bread crumb trail providing links back to each previous page that you navigated through in order to get to the current page. 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.

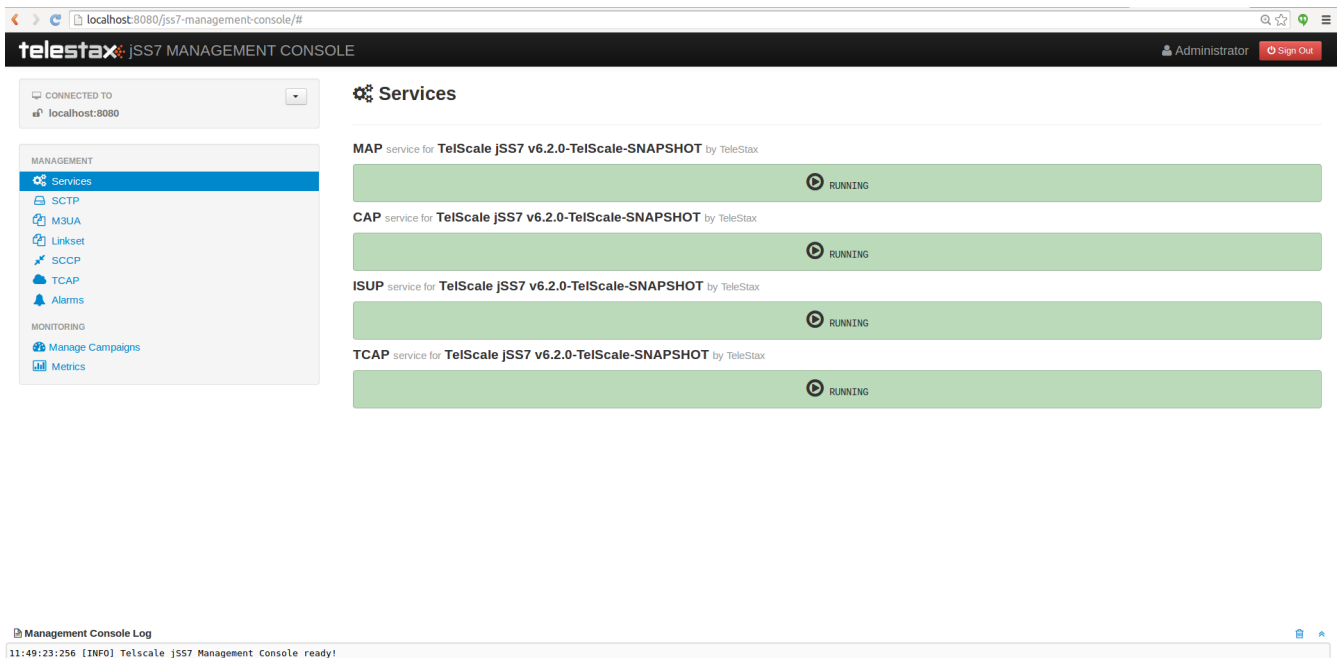


Figure 1. GUI - Services

The main window will display all the configured services and their current state (running or stopped). In the example above MAP, CAP, TCAP and ISUP are configured and running. If any of these services is not running currently, the page will indicate that service as 'Stopped'. This page only indicates the current state of a service and does not allow you to start or stop the service dynamically. You can view this screen anytime by clicking on the 'Services' link in the left panel.



You will notice bread crumbs at the top of the main panel, allowing you to return to any of the previous pages you navigated through.

## 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.

## Authentication

Restcomm jSS7 GUI Management Security is based on the JBoss Security Framework. This is explained in [\[security\]](#).

## Running Restcomm jSS7 Simulator

The Restcomm jSS7 comes with a Simulator module that will enable you to test and understand the functionality of the Stack. You can install and run the Simulator module on any machine. In addition, you can run several instances of the Simulator from one folder and each of these instances can have it's own configuration options. Configuration options are saved into a xml configuration file.

Each running instance of the Simulator has its own name (called "host name") and the name of

configuration file depends on this "host name". For example if the host name is "a1", then the name of the configuration file will be "a1\_simulator.xml". You must ensure that you provide each running Simulator instance with a unique host name in order to allow each running Simulator instance to have different configuration options.

You can run and manage the Simulator locally or remotely. For running the Simulator locally you will be using a GUI interface. If you are intending to run the Simulator remotely, then you can do so using RMI access (via a GUI interface) or a HTML interface (using a HTML Browser).

## Running the Simulator Locally

### *Procedure: Launching the Simulator Locally*

1. Pre-requisite: You must have SS7 Simulator installed locally in the machine.
2. Change the working directory to the bin folder in the Simulator's installation directory.

```
downloads]$ cd restcomm-jss7-<version>/ss7/restcomm-ss7-simulator/bin
```

3. (Optional) Ensure that the *run.sh* script is executable.

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

4. Execute the *run.sh* Bourne shell script with the command `./run.sh gui`. If you wish to pass the "host name" parameter (say for example "a1") you can do so by issuing the command as `./run.sh gui -na1` or `./run.sh gui --name=a1` where "a1" is the "host name" of this simulator instance.

```
bin$ ./run.sh gui --name=a1
```

5. Result: This will launch the GUI application form "Connecting to a testerHost ..." as in the figure below:

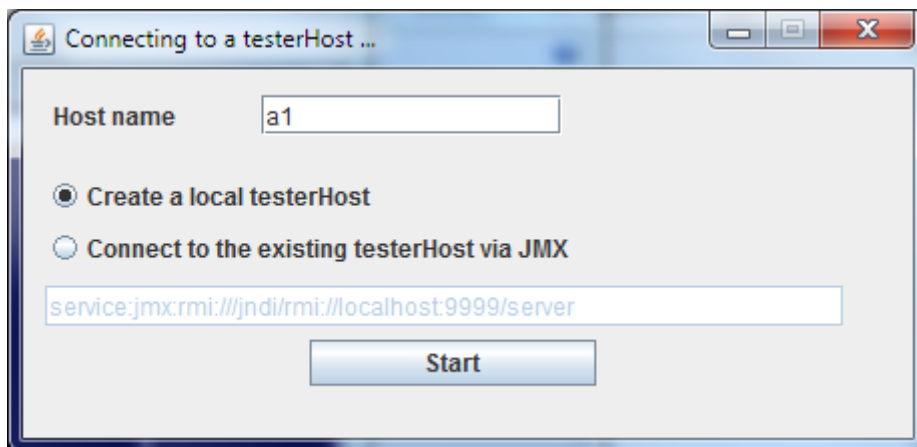


Figure 2. Running Simulator Locally

### *Procedure: Creating a Local testerHost*

1. Pre-requisite: You must have the GUI application form "Connecting to a testerHost" launched by running the simulator locally as explained in the above procedure.
2. In the GUI application form "Connecting to a testerHost ..." enter a value for the Host name field if it is empty.
3. Select the option "Create a local testerHost" and press the "Start" button.
4. Result: This will create a local testerHost and launch a new form "SS7 Simulator: <host name>-local" as in the figure below:

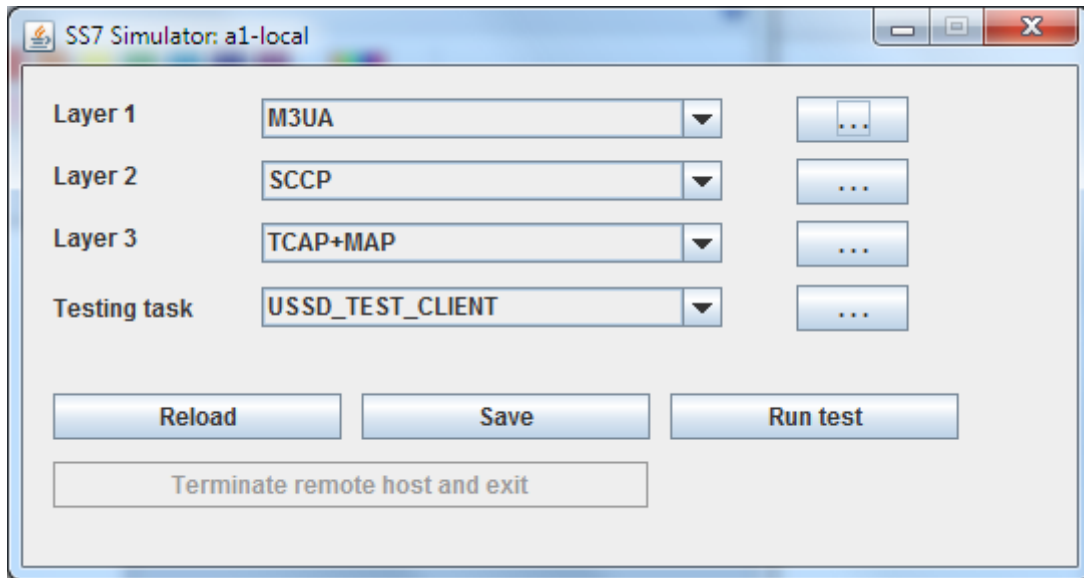


Figure 3. The main form of the Simulator

#### *Procedure: Running a Test*

1. Pre-requisite: You must have the main Simulator form launched as explained in the above procedure.
2. Select required modules for Layers 1 - 3 and the Testing Task.
3. Configure the modules to meet your requirements. Click on the button "..." to the right of the selected module. When layer configuring you can press one of two buttons "Load default values for side A" and "Load default values for side B" for loading default values for testing. These default values can be used if you are using SS7 Simulators in the one computer host as the side A and the side B for interaction. To save the configured layers to the disk, click the "Save" button. To reload the saved configuration, use the "Reload" button.
4. When all required layers are correctly configured, click the "Run test" button.
5. Result: The form for testing will be displayed as in the figure below:

**L1 state** SCTP: Connected M3UA: pFsm:ACTIVE IFsmP:ACTIVE

**L2 state** SCCP: Rspc: Enabled Rss: Enabled

**L3 state** TCAP+MAP: Started

**Testing state** TestUssdClient: CurDialog=No  
Count: processUnstructuredSSRequest-1, processUnstructuredSSResponse-1  
unstructuredSSRequest-0, unstructuredSSResponse-0, unstructuredSSNotify-0

Start Stop Refresh state

TimeStamp	Source	Message	UserData
1337664557...	SS7Event-TestUssd...	Rcvd: procUnstrSsResp: Balance=10\$	dialogId=1 DataCodingSchema=15
1337664555...	SS7Event-TestUssd...	Sent: procUnstrSsReq: *123#	dialogId=1 DataCodingSchema=15
1337664544...	SS7Event-M3UA	M3ua connection is active	Ass_a1
1337664543...	SS7Event-M3UA	Sctp connection is up	Ass_a1
1337664533...	SS7Event-TestUssd...	USSD Client has been started	
1337664533...	SS7Event-MAP	TCAP+MAP has been started	
1337664533...	SS7Event-SCCP	SCCP has been started	
1337664533...	SS7Event-M3UA	M3UA has been started	

**Message text** \*123#

Send ProcessUnstructuredRequest

Send UnstructuredResponse

Close current Dialog

**Operation result** ProcessUnstructuredSSRequest has been sent

**Message received** procUnstrSsResp: Balance=10\$  
PrevDialog: Sent procUnstrSsReq="\*123#";procUnstrSsResp="Balance=10\$";

Figure 4. Example Simulator testing form

The form has three sections:

- The top section displays information regarding the Simulator layers and the state of Testing. This information is refreshed every 5 seconds automatically. You can also choose to refresh this manually by clicking the "Refresh state" button. When you click the "Start" button, all modules will start and be ready for testing. When you click the "Stop" button all modules will stop.
- The middle section of the form displays notifications from testing modules.
- The bottom section of the form holds test dependant information whose details are displayed in the corresponding test definition.

For more details on executing tests, please refer to [\[simulator\\_tests\]](#).

## Running SS7 Simulator remotely

The Simulator can be run remotely using RMI and/or HTML Adaptors. But prior to launching the Simulator remotely, you must first run a tester host in the host machine as explained in the procedure below:

*Procedure: Launching a Tester Host in the host machine*

1. Pre-requisite: You must have SS7 Simulator installed locally in the host machine.
2. Change the working directory to the bin folder in the Simulator's installation directory.

```
downloads]$ cd restcomm-jss7-<version>/ss7/restcomm-ss7-simulator/bin
```

3. (Optional) Ensure that the *run.sh* start script is executable.

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

4. Execute the *run.sh* Bourne shell script with the command `./run.sh core -na1 -t8001 -r9999,9998` or `./run.sh core --name=a1 --http=8001 --rmi=9999,9998`. You can use a *rmi* connector or *http* connector or both simultaneously. The command options are explained below:

- "-n" (or "--name=") defines a host name (in the example above: "a1").
- "-r" (or "--rmi=") defines a port's listening for rmi requests (in the example above: "9999 and 9998"). RMI protocol uses 2 ports for server access. The first port (9999 above) is rmi port and 9998 is second port used by RMI. Usually RMI randomly selects this 2nd port, however if the core simulator is behind firewall, the GUI will not be able to connect from remote machine without admin opening up all the ports. This can be risky, hence --rmi takes two ports as comma separated and admin can open these two ports on firewall.



Please note that fixing the 2nd port in RMI contradicts the RMI specification.

- "-t" (or "--http=") defines a port listening for html requests (in the example above: "8001").
- "-Djava.rmi.server.hostname=192.168.1.1" - if remote host contains several IP addresses you need to specify by this command which exactly IP address RMI will use.

```
bin$ ./run.sh core -na1 -t8001 -r9999,9998
```

or

```
bin$ ./run.sh core --name=a1 --http=8001 --rmi=9999,9998
```

5. Result: This will start the Tester Host in this machine and the output displayed in the console will be as below:

```

=====

SS7 Simulator Bootstrap Environment

SIMULATOR_HOME: /home/vinu/restcomm-jss7-6.1.3.GA/ss7/restcomm-ss7-simulator

JAVA: /usr/lib/jvm/jre-1.7.0-openjdk/bin/java

JAVA_OPTS: -Dprogram.name=run.sh -Djava.net.preferIPv4Stack=true -Xms256m
-Xmx512m -Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000

CLASSPATH: /home/vinu/restcomm-jss7-6.1.3.GA/ss7/restcomm-ss7-
simulator/bin/run.jar

=====
.
.
.
.
.
All beans have been loaded...
RMI connector initializing...
RMI connector has been started...
Html connector initializing...
Html connector has been started...
Waiting for commands...

```

When the Tester Host is running successfully in the host machine, you can now run the Simulator tests remotely from any machine which has the Simulator installed. The following procedures provide instructions to run the Simulator remotely using the two interfaces (RMI and HTML).

*Procedure: Managing the Simulator Remotely using RMI Interface*

1. Pre-Requisite: The Tester Host must be launched in the host machine as explained in the above procedure. The local machine must have the Simulator installed.
2. The best client for managing the Simulator via a RMI interface is the GUI interface of Simulator. It can be launched in the same way as launching a Simulator locally:

```
./run.sh gui
```

This will launch the GUI form "Connecting to a testerHost...".



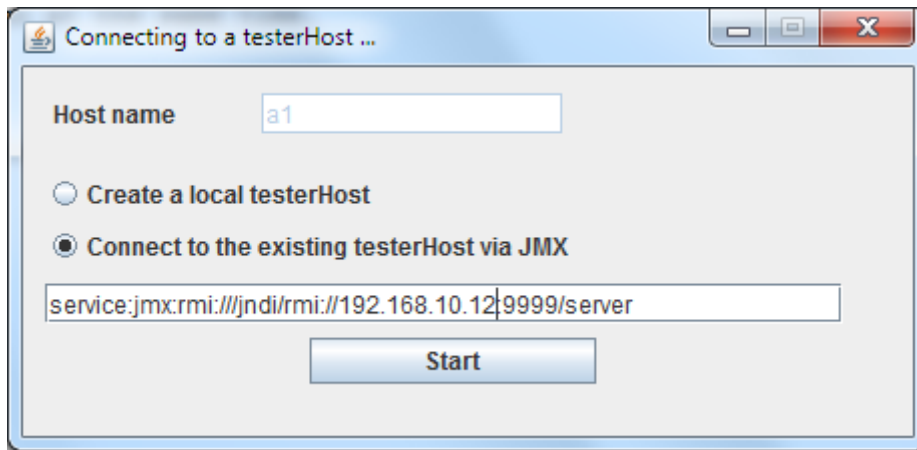


Figure 5. Launching Simulator GUI for a remote mode

3. In the form, select the option "Connect to the existing testerHost via JMX".
4. In the default RMI connection string, replace "localhost" with the correct IP address of the tester host computer and the value "9999" with the correct RMI port (the port from option "-r" or "--rmi=" as provided when launching the testerHost in the host machine).
5. Once successfully connected, the management will be similar to the procedures explained for running the Simulator application locally (see [Running the Simulator Locally](#)). "Jconsole" application can be also used as the client. But this application is less convenient when compared to the easy RMI interface.

*Procedure: Managing the Simulator Remotely using HTML Interface*

1. HTML management is less convenient than RMI management. But it can be used if RMI is not acceptable (for example if you are behind a proxy). For configuring the parameters and running the tests you can use any HTML browser. In the URL field of the browser, type "http://<IP address>:<port>" where IP address is the host machine's address and port is the value specified for "-t" or "--http=" option when launching the Tester Host in the remote machine.

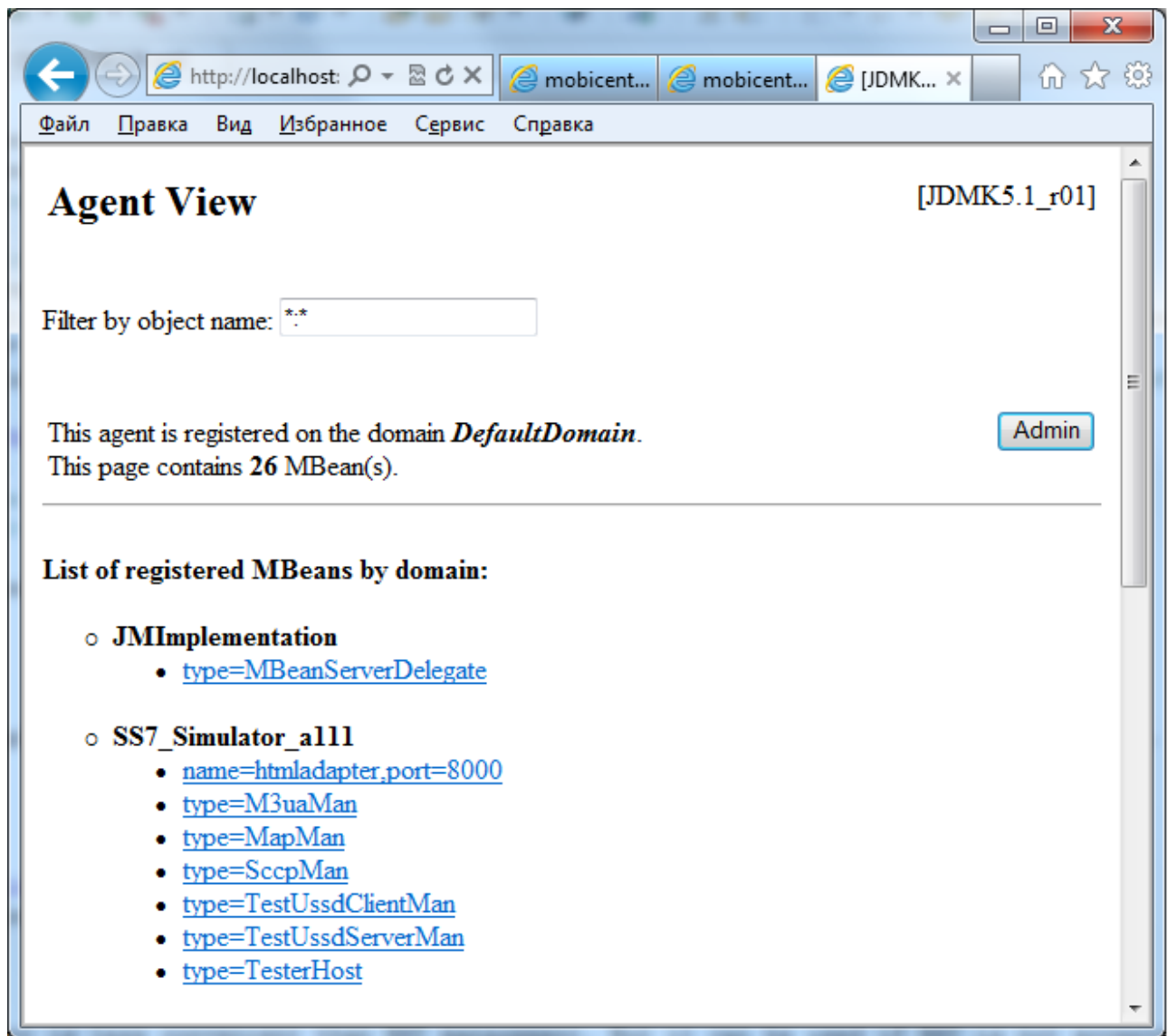
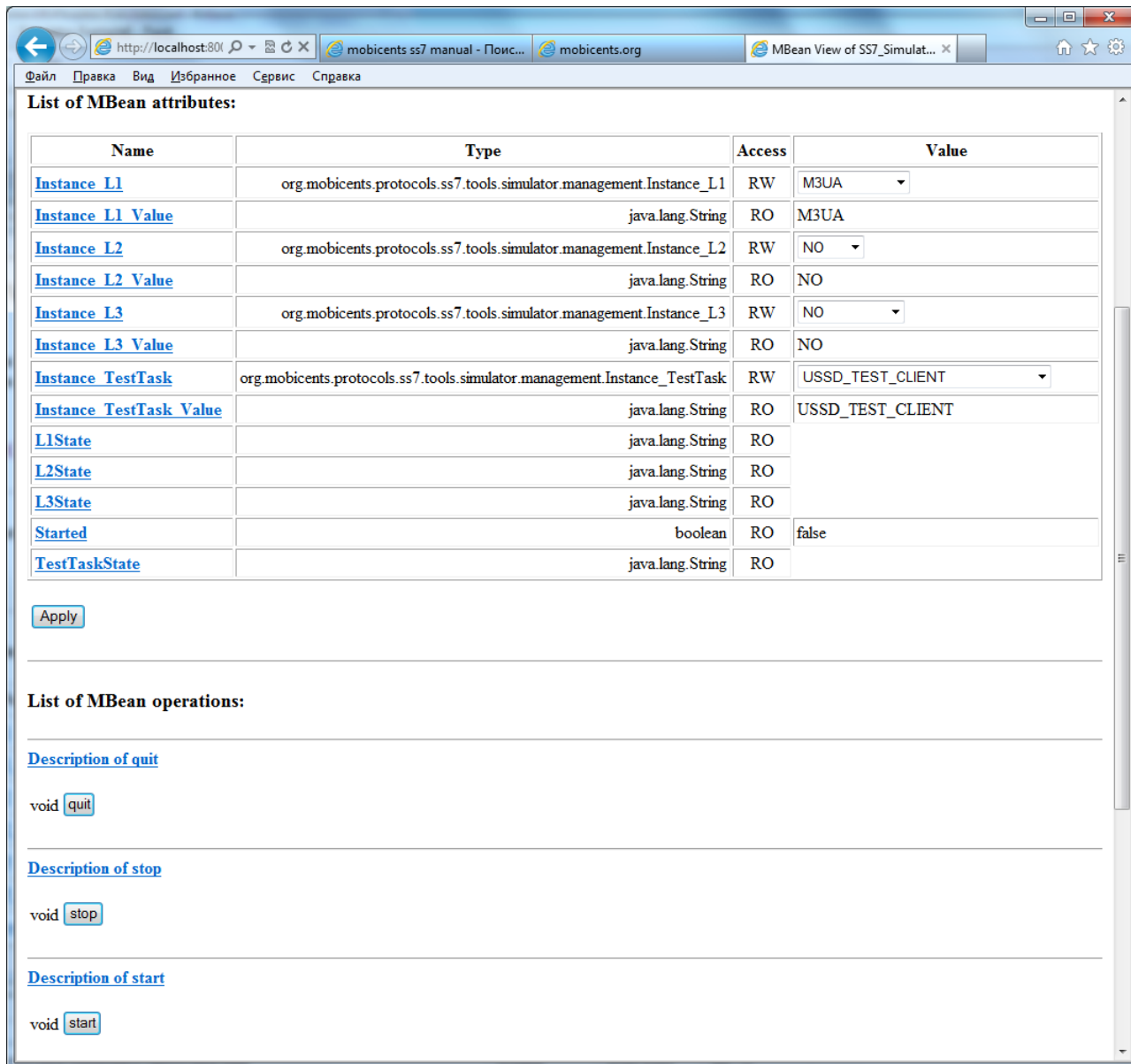


Figure 6. Managing remotely with a HTML Interface

2. The web page will display all Beans. "TesterHost" is the main MBean in which you can select the mode for test working, start/stop testing (buttons "Start"/"Stop") and quit a tester host (button "quit"). You can use other Beans for setting different options for layers and test tasks.



**List of MBean attributes:**

Name	Type	Access	Value
<a href="#">Instance L1</a>	org.mobicens.protocols.ss7.tools.simulator.management.Instance_L1	RW	M3UA
<a href="#">Instance L1 Value</a>	java.lang.String	RO	M3UA
<a href="#">Instance L2</a>	org.mobicens.protocols.ss7.tools.simulator.management.Instance_L2	RW	NO
<a href="#">Instance L2 Value</a>	java.lang.String	RO	NO
<a href="#">Instance L3</a>	org.mobicens.protocols.ss7.tools.simulator.management.Instance_L3	RW	NO
<a href="#">Instance L3 Value</a>	java.lang.String	RO	NO
<a href="#">Instance TestTask</a>	org.mobicens.protocols.ss7.tools.simulator.management.Instance_TestTask	RW	USSD_TEST_CLIENT
<a href="#">Instance TestTask Value</a>	java.lang.String	RO	USSD_TEST_CLIENT
<a href="#">L1State</a>	java.lang.String	RO	
<a href="#">L2State</a>	java.lang.String	RO	
<a href="#">L3State</a>	java.lang.String	RO	
<a href="#">Started</a>	boolean	RO	false
<a href="#">TestTaskState</a>	java.lang.String	RO	

**List of MBean operations:**

[Description of quit](#)

void

[Description of stop](#)

void

[Description of start](#)

void

Figure 7. TesterHost Bean

- The Results of the tests can be found at the console (in the server) or in the log file (file name is "a1.log" where "a1" is the name of the Simulator instance).