SMPPSIM Selenium Testing Tool

Introduction:

SMPPSim is a testing utility which mimics the behavior of an SMPP based SMSC. SMPP stands for Short Message Peer to Peer Protocol. SMSCs (Short Message Service Centers) from Logica Aldiscon support this protocol, as a means of allowing external (to the GSM network) applications to submit and receive SMS messages to / from the SMSC. Some other SMSC vendors also support SMPP.

If you are intending to implement support for SMS text messaging using the SMPP protocol in your application, you will need some way of testing your application. Ultimately of course, you should aim to use a real SMSC for testing before going live. Hopefully the network operator you are working with can supply a test SMSC for this purpose. In some cases however, two factors make SMPPSIM a useful tool for the initial stages of testing:

1. It can take months to acquire a connection to a GSM network
2. Some network operators run stringent acceptance tests before they will allow your application to interact with the production SMSCs. It can take weeks to complete these tests, even if things go well.

So, it pays to have some independent means of doing your initial testing. Firstly so that your project is not completely dependent on acquiring a network connection in order to be able to progress, and secondly so that your code is as good as you can get it before entering the formal acceptance testing phase with the network operator.

How to Setup SMPPSIM Utility:

You need to setup SMPPSIM on your system properly before actually using it for your testing. Please follow the following steps to setup it on your machine:

1. Clone the SMPPSIM Repository on machine where your application is deployed that you want to testfrom the following Cloning URL:
   'git clone https://bilalahmed54@bitbucket.org/expertflow-ondemand/smppsim.git'

   Furthermore, you can acquire SMPPSIM utility from bitbucket:
   https://bitbucket.org/expertflow-ondemand/smppsim

2. Locate the configuration file named as 'smppsim.props'. You can find this file under conf directory of cloned SMPPSIM Repository.

3. Open this file and find the OUTBIND related attributes. Such as:

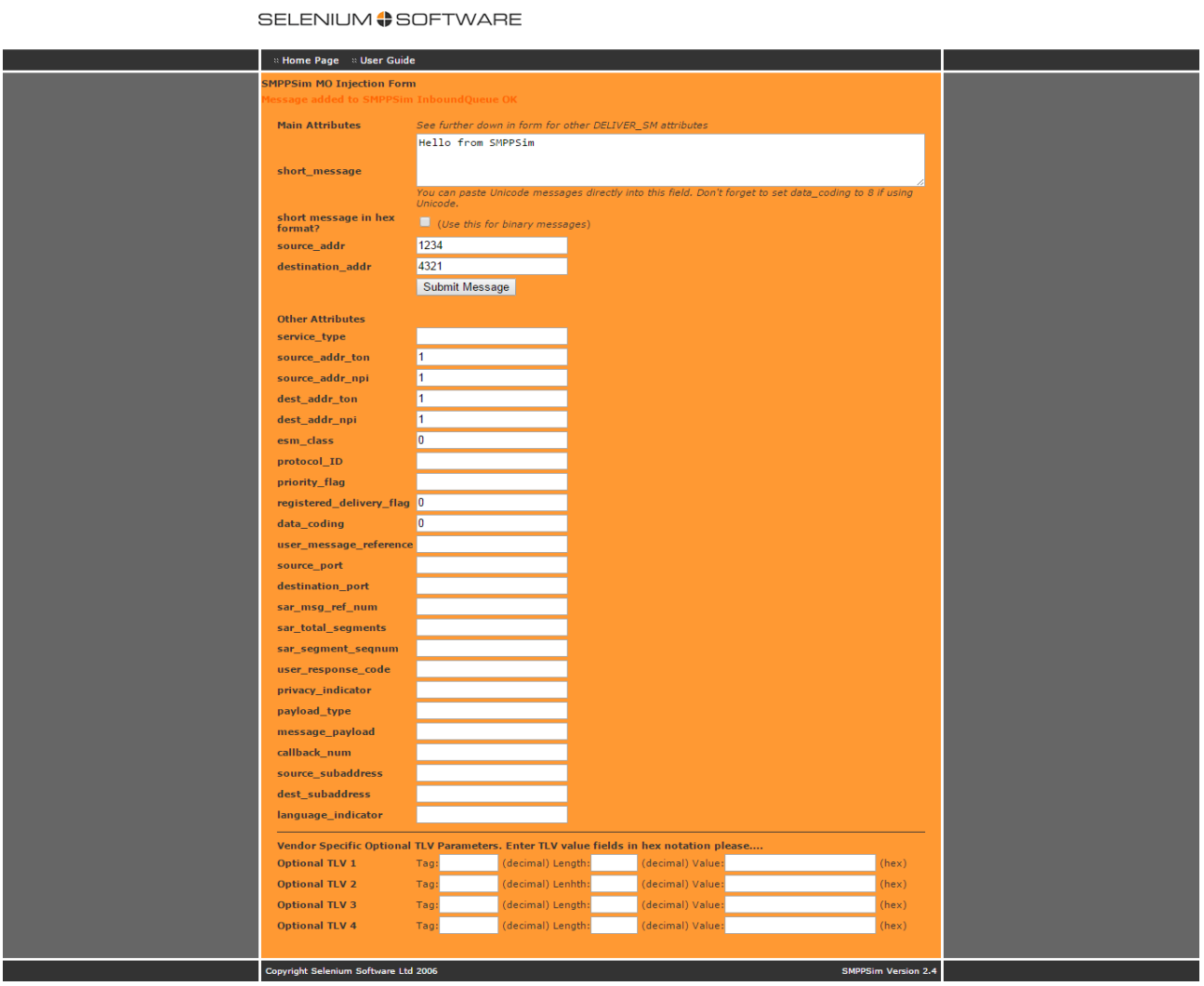
   SYSTEM_IDS=smppclient1,smppclient2
   PASSWORDS=password,password
   OUTBIND_ENABLED=false
   OUTBIND_ESME_IP_ADDRESS=127.0.0.1
   OUTBIND_ESME_PORT=2776
   OUTBIND_ESME_SYSTEMID=smppclient1
   OUTBIND_ESME_PASSWORD=password

   These attributes will be used to set application SMPP Settings in the following steps.

4. Update your application SMPP related settings in Application in Code or in Interface (if you have any) with the above attributes. Now, you SMPP Settings should look like this:

5. Locate the configuration file named as ‘startsmppsim.bat’. You can find this file under cloned SMPPSIM Repository. Just run this batch file with double click on it and keep the command prompt open.
   Note: If you don’t want to run it as command prompt as you will have to keep an additional window open you can also run it as window service in background.
6. After Updating these SMPP settings, restart gateway application so that the SMPP settings we changed may be applicable.

7. Access this URL: ‘http://localhost:88/inject_mo’. You will get window like this:



You can use this interface to send SMS using specified attributes such as message, source, and destination. So, we can send SMS to our application for testing purposes which will mimics the behavior when Customer sends SMS to any Service.

8. Now, Access URL: ‘http://localhost:88/?stats’. You will get window like this:

Source address: 4321
Destination address: 1234
This is Reply Text...!!!

Source address: 4321
Destination address: 1234
This is Reply Text...!!!

Source address: 4321
Destination address: 1234
This is Reply Text...!!!

So, in this Interface you will be seeing the Outbound SMS generated by our application. You will also be seeing Source Address that our application used to send SMS and also the destination number that to which number our application sent SMS.

9. And If you want to see the overall behavior such as total MOs generated, SMS delivered and error etc. then you can see using this URL: ‘http://localhost:88/’. You will be seeing window like this:

SELENIUM SOFTWARE

:: Refresh :: Inject an MO Message :: User Guide :: Shutdown SMPPSim

SMPPSim Control Panel

SMPPSim has been running since:	Tue, 2 Jun 2015 12:44:07
TX sessions bound	0
RX sessions bound	0
TRX sessions bound	1
InboundQueue Size (MO messages)	0
PendingQueue Size (MO messages, no session available)	1
OutboundQueue Size (MT messages)	3

PDU Statistics

PDU Type	ESME_ROK count	Error Count
BIND_TRANSMITTER	0	0
BIND_RECEIVER	0	0
BIND_TRANSCEIVER	1	0
OUTBIND	0	0
SUBMIT_SM	6	0
SUBMIT_MULTI	0	0
DELIVER_SM	8	0
DATA_SM	0	0
QUERY_SM	0	0
CANCEL_SM	0	0
REPLACE_SM	0	0
ENQUIRE_LINK	687	0
UNBIND	0	0
GENERIC_NAK	0	0

Copyright Selenium Software Ltd 2006 SMPPSim Version 2.4

So, in this window you can see the overall stats/behavior of the application.

Note: Regarding last three steps you are accessing Selenium SMPPSIM tool using URL starting with 'http://localhost:88'. In this URL, 'localhost' is the default IP and '88' is the default port on which machine you are running this tool. So, this IP and Port will change accordingly if you change it in SMPPSIM properties file 'smppsim.props'. Furthermore, if you want to change default port, you can change from this properties file.

If you face any issue while setting up this tool or you have any query regarding this. You can contact me using this email address: **bilal.ahmed@expertflow.com**