SMPP Acceptance Test Plan

Version 10

For connection to

Connection Software's

SMPP servers

1 Contents

2 Glossary

ESME	External Short Message Entity – generally Connection Software's customer's system
PDU	Protocol Description Units
SMPP	Short Message Peer to Peer
SMSC	Short Message Service Centre – generally the Connection Software systems
Stability Period	A period, generally of 1 week,

3 Connection Software's SMPP service

The current API supports the following SMPP version 3.3 and 3.4 Protocol Description Units (PDU's):

- BIND_TRANSMITTER / BIND_TRANSMITTER_RESP
- BIND_RECEIVER / BIND_RECEIVER_RESP
- SUBMIT_SM / SUBMIT_SM_RESP
- ENQUIRE_LINK / ENQUIRE_LINK_RESP
- QUERY_SM / QUERY_SM_RESP
- DELIVER_SM / DELIVER_SM_RESP
- BIND_TRANCEIVER / BIND_TRANCEIVER_RESP
- UNBIND / UNBIND_RESP

You must honour the unbind PDU by issuing an *unbind_resp* PDU before disconnecting.

You are also required to honour the *enquire_link* PDU by sending an *enquire_link_resp* PDU.

3.1 Binding

Customers should bind as either a Transmitter or a Transceiver.

Customers may wish to bind as a Receiver if they are bound as a Transmitter and wish to receive Delivery Receipts or MO traffic.

Port 2775 on both s3.itsarrived.net and s4.itsarrived.net are open for SMPP connections. You should bind to both servers as either one may be taken off-line for maintenance. Binding to both will greatly enhance your service reliability.

When you are using our 2-way MO service it is **essential** that you are bound in either as a receiver or a transceiver to both servers.

The following credentials should be used in all bind PDU's:

- system_id – your username
- password – your PIN
- system_type – blank

3.2 Submitting messages

Our SMSC default alphabet (data_coding=0x00) is the GSM 03.38 7-bit ASCII alphabet.

We fully support the following character sets:

- data_coding = 0x01 – IA5 ASCII ANSI X3.4
- data_coding = 0x03 - ISO-8859-1 (ISO Latin 1)
- data_coding = 0x05 - JIS
- data_coding = 0x06 - ISO-8859-5 (Cyrillic)
- data_coding = 0x07 - ISO-8859-8 (Latin/Hebrew)
- data_coding = 0x08 – UCS2 ISO/IEC 10646

We support the SMPP version 3.4 message_payload (tag ID 0x0424) optional parameter. This allows you to send us messages that are longer than160 characters. We will split long messages before passing them to Network Operators.

The message ID is returned in the *sm_submit_resp* PDU as an 8-digit hexadecimal number.

3.3 TON and NPI

Please note that at present we ignore the *dest_addr_ton* and *dest_addr_npi* settings that are supplied since there is often difficulty in choosing the correct values instead we determine appropriate values based on the Originator Address supplied.

3.4 Receipts

To receive delivery receipts, you need to configure SMPP as the delivery mechanism for delivery receipts on your account. You can do this via the website: log into your account and select "Configure Services" => "Delivery Receipts Self-Provisioning". This will route all delivery receipts for messages sent on your account to your SMPP Receiver or Transceiver bind. The format of delivery receipts is as given in Appendix B of the SMPP v3.4 specification.

The message ID in the delivery receipt is given as a 10-digit decimal number.

If your SMPP Receiver or Transceiver bind is unavailable we will try to resend receipts for up to 24 hours.

3.5 Error handling

The SMPP API supports the SMPP *command_status* as specified in section 5.1.3 of the SMPP Protocol Specification v3.4

In addition to the standard command_status values it is important that you expect and deal correctly with the command_status 0x00000400 – ESME_RNOCREDIT. This code will be returned when you are out of credit.

- If you receive ESME_RNOCREDIT in response to a bind PDU you should NOT try to rebind into either server until you have sufficient funds in your account.
- If you receive ESME_RNOCREDIT as a response to a submit PDU you should NOT try to resend the message to either server. An unbind PDU will be issued by the SMSC and this must be honoured. You may attempt to rebind provided you do so in conformance with the Standoff Policy below.

3.6 Resilience

ENQUIRE_LINK should be set to 60 seconds.

3.7 Standoff Policy

Never try to retry *submit_sm* PDU's that have been refused for a genuine reason like

- invalid number [ESME_RINVDSTADR (0x0000000B)] or
- out of credit [ESME_RNOCREDIT (0x00000400)]

If your Bind or Submit PDU is refused and you wish to retry it, please adhere to our stand-off strategy.

You should wait for 30 seconds for each of a maximum of 3 attempts after which you should wait for 5 minutes before resending the PDU.

3.8 Mobile numbers for testing

The number range 07700 900000 to 900999 is designated by Ofcom for Drama purposes i.e. for Radio and Television. This range of 1000 numbers can be used for testing and no messages will be sent or charges incurred.

Note:

44 7700 900006 will always return an Invalid Number status ESME_RINVDSTADR

44 7700 900015 will always return an Out of Credit status ESME_RNOCREDIT

3.9 Systems for testing

For testing only, the primary bind should be to a test system whose IP address will be advised. Access to the test system will be provided on request.

4 Acceptance Test Overview

This test plan has been produced to validate the compliance of customer's ESME with Connection Software's requirements for operational stability. Such compliance will normally be established prior to the connection of such ESME to a live SMSC in the Connection Software network.

Please note that throughout the document the terms 'ESME' and 'ESME' are used interchangeably.

This test plan document is split into 4 sections:

4.1 SMPP Protocol Tests

These tests will ensure that the ESME is able to send/receive all required SMPP primitives correctly to/from the SMSC.

4.2 Performance Tests

These tests will ensure that the ESME behaves correctly during failure conditions on the link to the SMSC, and operates correctly on the detection of error conditions.

It is strongly recommended that the ESME maintains a Log file of detected errors & abnormal conditions.

Many of these tests may be performed in more than one way. It is often easier to verify ESME functionality by connecting to a dummy SMSC Test Program rather than a real SMSC. For this reason the term 'SMSC' in this section can be taken to mean a real SMSC, or a test program imitating an SMSC.

4.3 Resilience Tests

These tests will ensure that the ESME can successfully switch to Connection Software's secondary SMSC in the event of the primary SMSC being unavailable.

4.4 Recording results

All results from these tests should be documented at the time of testing and reviewed before the start of the Stability Period. Any non-conformity should be fully resolved before entering the Stability Period.

4.5 Stability Criteria

This section gives the criteria which an ESME should fulfil to ensure stability of the application-generated SMS traffic and hence the entire end-to-end system.

The ESME should first pass the SMPP Protocol, Performance and Resilience tests and then enter a Stability Period. All ESME are expected to undergo the full Stability Period

At the end of the agreed Stability Period the ESME should be verified against the Stability Criteria.

If the Stability Criteria are met then the system can be considered fit for use.

The Stability Criteria should always be tested.

Other tests in the above sections are only required for each individual operation that the ESME involved actually intends to use. Thus if a particular application does not use Long Messages then test 6.6 is not required.

5 Test planning

The following checklist should be used to determine record which tests are required.

The Stability Criteria should always be met.

Other tests may or may not be appropriate to the needs of the customer and the Acceptance Test should be planned to test all of the features that the customer knows or anticipates that they will need.

Test results should be recorded on this document.

Test number	Description	To be tested ?	Result
6	Protocol Tests		
6.1	Bind to the SMSC as a Transmitter or Transceiver	**Y**	
6.2	Bind to the SMSC as a Receiver		
6.3	Verify that the ESME supports the GSM 03.38 character set		
6.4	Verify that the ESME supports the ISO 8859-1 Latin1 character set		
6.5	Verify that the ESME supports UCS2		
6.6	Submit a Concatenated Message		
6.7	Submit a Long Message		
6.8	Submit a Message with Dynamic TPOA		
6.9	Submit a Message to an invalid number	**Y**	
6.10	Unbind a Transmitter or Transceiver from the SMSC	**Y**	
6.11	Unbind a Receiver from the SMSC		
7	Performance tests		
7.1	ENQUIRE_LINK	**Y**	
7.2	Reconnect on underlying data link closure	**Y**	
7.3	Response of ESME to an SMSC ESME_RNOCREDIT status	**Y**	
7.4	Standoff in conjunction with ESME_RNOCREDIT	**Y**	
8	Resilience test		
8.1	Response of ESME connected to two SMSCs to an SMSC going down	**Y**	
9	Stability criteria		
9.1	Traffic conformance with the SLA	**Y**	
9.2	Dummy traffic is not generated	**Y**	
9.3	Messages do not queue up the SMSC	**Y**	
9.4	ESME does not continuously bind/unbind or failover	**Y**	

Test marked “Y” should be passed before live traffic is passed to Connection Software’s production servers to ensure that Quality of Service is maintained.

6 Protocol Tests

6.1 Bind to the SMSC as a Transmitter or Transceiver

Objective	• The ESME binds successfully to the SMSC as a Transmitter or Transceiver • Verify that the bind transmitter syntax complies with the SMPP specification
Initial Conditions	• TCP connectivity has already been tested to SMSC • The SMSC has been correctly configured with the application's details
Execution	• The SMPP customer initiates a bind transmitter from their application. It is acceptable for: • this to be initiated by an MO message if required or • The TX Bind request to only be attempted when there are messages to be sent if this is how the application has been designed. • Connection Software checks on the SMSC that the SMPP client had successfully bound to the SMSC • Connection Software will verify that the bind transmitter PDU complies with the SMPP specification
Expected Results	• A bind transmitter is received by the SMSC • The SMSC returns a bind transmitter response
Actual Results	

6.2 Bind to the SMSC as a Receiver

Objective	• Using the ESME, bind successfully to the AIM on the SMSC as a receiver • Verify that the bind receiver syntax complies with the SMPP specification.
Initial Conditions	• Connectivity has been tested to SMSC • The SMSC has been correctly configured with the application's details
Execution	• The SMPP customer initiates a bind Receiver from their application • Connection Software checks on the SMSC that the SMPP client had successfully bound to the SMSC • Connection Software will verify that the bind transmitter PDU complies with the SMPP specification
Expected Results	• A bind receiver is received by the SMSC • The SMSC returns a bind transmitter response
Actual Results	

6.3 Verify that the ESME supports the ISO 8859-1 Latin1 character set

Objective	• To verify that the ESME encodes text messages in ISO 8859-1
Initial Conditions	• The ESME is set up and correctly configured on the SMSC • ESME is bound in as a receiver • TCP trace is running on Connection Software's SMSC
Execution	• From the ESME send an Submit_SM to the SMSC with the following English text: **ISO 8859-1 £100 # $ e.mail@address** • Connection Software to verify that in the Submit_SM the pound sign is represented by **A3** hex, the # sign by 23 hex, the **$** by hex 24 and the @ by **40** hex • Verify that all the characters are successfully delivered to the handset
Expected Results	▪ The ESME encodes characters in ISO 8859-1 before sending them to the SMSC
Actual Results	

6.4 Verify that the ESME supports UCS2

Objective	• To verify that the ESME encodes text messages in UCS2
Initial Conditions	• The ESME is set up and correctly configured on the SMSC • ESME is bound in as a receiver • TCP trace is running on Connection Software's SMSC
Execution	• From the ESME send an Submit_SM to the SMSC with the following Arabic text: ا ب ت • These characters can be entered on a Windows keyboard by holding down the ALT key and pressing 4 numbers on the numeric keypad – as follows • Aleph ا <ALT>+1575 • Ba ب <ALT>+1576 • Ta ت <ALT>+1578 • Connection Software to verify that in the Submit_SM the aleph character ا is represented by hex 0627, ba by hex 0628 and ta by hex 062A.
Expected Results	▪ The ESME encodes characters in UCS2 before sending them to the SMSC
Actual Results	

6.5 Submit a Concatenated Message

Objective	• To verify that the ESME can send a concatenated message to the handset
Initial Conditions	• The ESME is set up and correctly configured • The SMSC has been correctly configured with the application details • ESME is bound in as a transmitter and receiver • TCP trace is running on Connection Software's SMSC
Execution	• ESME sends a concatenated message to the handset with the registered_delivery_flag set • Connection Software to confirm that the SMPP UDHI flag is set, i.e. the ESM class is 0x40 • Connection Software to confirm that source address is in correct format • Connection Software to confirm that data coding scheme (DCS) is set to the correct value • Connection Software to confirm that protocol identifier (PID) is set to either 0 or 3F • Connection Software to verify that a delivery receipt is generated, sent to the ESME and successfully acknowledged
Expected Results	• A concatenated message is submitted to the SMSC and forwarded to handset • Message submitted with SMPP UDHI flag set correctly • Message submitted with correct DCS and a PID value of either 0 or 3F • Message has correct source address value • The delivery receipt generated by the SMSC is correctly acknowledged by the ESME
Actual Results	.

As an alternative the ESME may send a long message directly.

6.6 Submit a Long Message

Objective	• To verify that the ESME can send a concatenated message to the handset
Initial Conditions	• The ESME is set up and correctly configured • The SMSC has been correctly configured with the application details • ESME is bound in as a transmitter and receiver • TCP trace is running on Connection Software's SMSC
Execution	• ESME sends the following long message to the handset "The Quick Brown Fox Jumped Over The Lazy Dog's Back 1,234,567,890 times and forsaking monastic tradition, twelve jovial friars gave up their vocation for a questionable existence on the flying trapeze." • Connection Software to confirm that the SMPP sm_lenght is zero and SHORT_MESAGE is empty • Connection Software to confirm that the SMPP message_payload contains the message. • Connection Software to confirm that the SMPP UDHI flag is not set, i.e. the ESM class is 0x00 • Connection Software to confirm that source address is in correct format • Connection Software to confirm that data coding scheme (DCS) is set to the correct value • Connection Software to confirm that protocol identifier (PID) is set to either 0 or 3F
Expected Results	• A long message is submitted to the SMSC and forwarded to handset • Message submitted with SMPP UDHI flag set correctly • Message submitted with correct DCS and a PID value of either 0 or 3F • Message has correct source address value • The delivery receipt generated by the SMSC is correctly acknowledged by the ESME
Actual Results	

6.7 Submit a Message with Dynamic TPOA MSISDN

Objective	• To verify that the ESME can submit text messages to the SMSC
Initial Conditions	• The ESME is set up and correctly configured • The SMSC has been correctly configured with the application's details • ESME is bound in as a transmitter • TCP trace is running on Connection Software's SMSC
Execution	• The ESME sends a "Submit_SM" to the SMSC using a MSISDN supplied by Connection Software e.g. 447700900987 • Connection Software to confirm that destination address in Submit_SM is in international format, i.e. the MSISDN begins with a country code, (e.g.: 447802) • Connection Software to confirm that source address is in correct format • Connection Software to confirm that data coding scheme (DCS) is set to 0 • Connection Software to confirm that protocol identifier (PID) is set to 0 • Connection Software to confirm that all parameters are checked to ensure that ESME has their application set correctly
Expected Results	• Message is submitted to SMSC with registered flag not set and forwarded to handset • Message submitted with correct DCS and PID values • Message has correct source address value
Actual Results	

6.8 Submit a Message to an invalid number

Objective	• To verify that the ESME cannot submit text messages using the barred code to the SMSC
Initial Conditions	• The ESME is set up and correctly configured • The SMSC has been correctly configured with the application's details • ESME is bound in as a transmitter • TCP trace is running on Connection Software's SMSC
Execution	• The ESME sends a "Submit_SM" to the SMSC using a MSISDN supplied by Connection Software e.g. 447700900006 • Connection Software to confirm that destination address in Submit_SM is in international format, i.e. the MSISDN begins with a country code, (e.g. 44) • Connection Software to confirm that source address is in correct format • Connection Software to confirm that the correct error code is returned.
Expected Results	• Message is submitted to SMSC with registered flag not set and forwarded to handset • The SMSC returns the corresponding error code.
Actual Results	

7 Performance tests

7.1 Unbind a Receiver from the SMSC

Objective	• To verify that the ESME issues an unbind receiver command when closing down the session
Initial Conditions	• The ESME is set up and correctly configured on the SMSC • ESME is bound in as a receiver • TCP trace is running on Connection Software's SMSC
Execution	• From the ESME send an unbind receiver command to the SMSC • Verify that the command is received and that the ESME has unbound the receiver
Expected Results	• An UNBIND_RESP message will be received from the SMSC
Actual Results	

7.2 ENQUIRE_LINK

Objective	• Prove that the ESME sends an ENQUIRE_LINK PDU every 60 seconds on all binds irrespective of other traffic being submitted.
Initial Conditions	• The ESME is set up and correctly configured on the SMSC • The SMSC is configured with an 80 seconds idle timeout period • TCP trace is running on Connection Software's SMSC
Execution	• Start ESME so that it binds to the SMSC as both a transmitter and receiver • Wait 2.5 minutes and verify that at least 2 ENQUIRE_LINK messages have been sent 60 seconds apart from the ESME to the SMSC bind session • Verify that the ESME does not drop any of the two connections to the SMSC
Expected Results	• The ESME generates an ENQUIRE_LINK PDU every 60 seconds on both the transmitter and receiver sessions
Actual Results	

7.3 Reconnect on underlying data link closure

Objective	• To verify that the ESME will re-connect successfully if the underlying data connection is suddenly closed • Throughout the test operation below it is expected that appropriate alarms will be displayed to the customer operator by the ESME
Initial Conditions	• The ESME is set up and correctly configured on the SMSC • Application is bound to SMSC as a transmitter and receiver • TCP trace is running on Connection Software's SMSC
Execution	• On the SMSC kill the UNIX processes associated with the ESME's transmitter and receiver session • Verify that ESME re-establishes a connection within 60 seconds
Expected Results	• The ESME re-establishes a transmitter and receiver connection to the SMSC within 60 seconds • The ESME generates an alarm
Actual Results	

7.4 Response of ESME to an SMSC ESME_RNOCREDIT status

Objective	• To test satisfactory operation of ESME when the SMSC reports that the customer is out of credit.
Initial Conditions	• The ESME is set up and correctly configured on the SMSC • Application is bound to SMSC as a transmitter and receiver • TCP trace is running on Connection Software's SMSC
Execution	• Connect to the SMSCs • Client will submit a message to +447700900015 on which the SMSC will generate an ESME_RNOCREDIT followed by UNBIND
Expected Results	• The ESME should not resend the message • The ESME should honour the unbind • The ESME should disconnect. • The ESME may try to reconnect according to our standoff policy.
Actual Results	

7.5 Standoff in conjunction with ESME_RNOCREDIT

Objective	• To test the ESME standoff capability in conjunction with the ESME response to an out of credit bind failure.
Initial Conditions	• Connectivity has been tested to SMSC • The SMSC has been correctly configured with the application's details
Execution	• Connect to the SMSC as a Transmitter or a Transceiver using username XXXX and password YYYY • SMSC will reject the bind request with ESME_RNOCREDIT • ESME should standoff for 30 seconds up to 3 times after which it will standoff for 5 minutes
Expected Results	• The ESME adheres to the standoff policy
Actual Results	

8 Resilience test

8.1 Response of ESME connected to two SMSCs to an SMSC going down

Objective	• To test satisfactory operation of ESME when a second SMSC (and possibly, link) is/are forced to be used for resilience, in the event of one going down.
Initial Conditions	• The ESME is set up and correctly configured on the SMSC • Application is bound to SMSC as a transmitter and receiver • TCP trace is running on Connection Software's SMSC
Execution	• Connect to both SMSCs simultaneously • Begin submission of 500 messages to the overall process. • Shut down the connection to one of the SMSCs, and observe consequences.
Expected Results	• The ESME successfully reroutes all remaining messages to the second SMSC
Actual Results	

9 Stability Period and Stability Criteria

Stability criteria are checked against the behaviour of traffic generated by the application during a Stability Period that will generally be 5 working days in duration.

During the Stability Period, which takes place immediately after the testing detailed in this Test Plan, the ESME is required to put a certain fraction of expected live traffic through the SMSC.

The decided percentage of live traffic is varied on a case by case basis according to the expected live service traffic & complexity of operations being performed. **As a general guideline during the Stability Period it is expected that at least 60 messages per hour are submitted on average.**

9.1 Traffic conformance with the SLA

The traffic generated by the ESME will conform to the volume limits specified in the SLA between Connection Software and the customer i.e. no more that 18000 messages per hour for a standard configuration.

Similarly the ESME should not generate more than 1000 *query_sm* requests per hour.

Advice may also be given by Connection Software if the number of message attempts (i.e. the actual number of message retries, including the first attempt, made by the SMSC for a single submission) proves to be excessive. This may happen, for instance, in ESME that are frequently sending messages to terminals that are often unavailable/out of coverage.

9.2 Dummy traffic is not generated

No messages will be submitted to the SMSC, by either the ESME or a service associated mobile terminal, that cause messages to be sent that have no valid destination address i.e. MSISDN

9.3 Messages do not queue up the SMSC

The message queue at the SMSC does not show a tendency to build up and never exceeds 200 messages.

9.4 ESME does not continuously bind/unbind or failover

The ESME should bind in once to each SMSC at the beginning of the Stability Period, and unbind once at the end. Binding/unbinding regularly or continuously is not allowed.

Traffic should normally be directed at the primary SMSC and the secondary SMSC should only be used for live traffic if the primary SMSC becomes unavailable.

The ESME must only retry in accordance with the Standoff Policy.

10 Contact information

Connection Software
391 City Road
London
EC1V 1NE
United Kingdom

Telephone	020 7713 8000	international	+44 20 7713 8000
Fax	020 7713 8001		+44 20 7713 8001
Email	support@csoft.co.uk		
Website	www.csoft.co.uk		