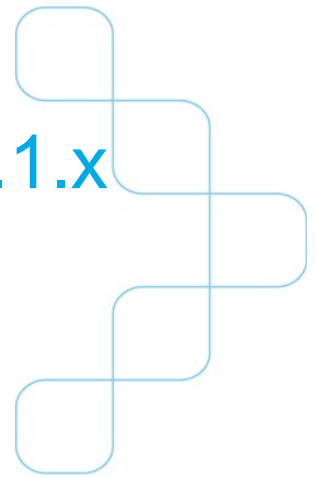


SMPP Resource Adaptor v2.1.x

Administration Guide

16 OCTOBER 2013



OpenCloud

Copyright and Disclaimers

Copyright © 2013 OpenCloud Limited. All rights reserved.

OpenCloud™ is a trademark of OpenCloud Limited.

Edinburgh House, St John's Innovation Park, Cowley Road, Cambridge CB4 0DS, United Kingdom

Documentation is provided "as is" and all express or implied conditions, representations and warranties, including any implied warranty of merchantability, fitness for a particular purpose or non infringement, are disclaimed, except to the extent that such disclaimers are held to be legally invalid.

The information in this document is confidential and meant for use only by the intended recipient and only in connection with and subject to the terms of its contractual relationship with OpenCloud.

Acceptance and/or use of any of the information contained in this document indicates agreement not to disclose or otherwise make available to any person who is not an employee of the intended recipient, or to any other entity, any of the information contained herein. This documentation has the sole purpose of providing information regarding OpenCloud software products and/or services and shall be disclosed only to those individuals who have a need to know.

Any entity or person with access to this information shall be subject to this confidentiality statement. No part of this publication may be reproduced or transmitted in any form or by any means for any purpose without the express written permission of OpenCloud.

Strictly Confidential

The information in this document is confidential and only available under OpenCloud's Non-disclosure Agreement (NDA). It must not be distributed or disclosed to any 3rd party. OpenCloud reserves the right to change the Product Portfolio information without notice or consultation.

No part of this publication may be reproduced or transmitted in any form or by any means for any purpose.

Contents

Administration Guide

1 Configuring the SMPP RA.....	2
2 Running the example SMPP RA services.....	4



 **OpenCloud**

OpenCloud's Rhino is a real-time application server for agile development, deployment and efficient management of person-to-person communication services across current and next generation technology. Rhino is a high performance, genuinely carrier-grade service execution environment for realizing a Next Generation Service Delivery Platform (NG-SDP). It uses commercial-off-the-shelf (COTS) hardware and software to deliver service layer agility to TDM and IP-based networks at a radically lower price-point than traditional solutions from network equipment providers.

OpenCloud headquarters are in Cambridge, United Kingdom. R&D, Engineering and Support are located in New Zealand, Spain and there are OpenCloud branch offices in the United States, Singapore and Japan.

For more information go to:

www.opencloud.com

<http://developer.opencloud.com>

SMPP Resource Adaptor Administration Guide



This document details basic procedures for system administrators deploying, managing and configuring the SMPP Resource Adaptor **v2.1.x**.

Topics

This document includes the following topics:

- [1 Configuring the SMPP RA](#) — configuration property descriptions and defaults for the SMPP and SMPP Bound resource adaptors
- [2 Running the example SMPP RA services](#) — how to install and run the example services included with the SMPP RA.

Audience and Scope

Intended Audience

This document is aimed at readers who:

- have a general, high-level understanding of telecommunication protocols
- are familiar with the OpenCloud SMPP Resource Adaptor
- want to install and administer the SMPP Resource Adaptor.

Scope

This document covers procedures for administering and deploying the SMPP Resource Adaptor.

This document does not focus on:

- SMPP Resource Adaptor APIs — see the [SMPP Generic](#) and [SMPP Bound](#) Resource Adaptor Javadocs.

1 Configuring the SMPP RA

SMPP Resource Adaptor configuration properties

Below are configuration properties for the [SMPP](#) and [SMPP Bound](#) resource adaptors.

SMPP RA configuration

Property	Type	Default	Description
<code>SmppListenAddress</code>	String	null (listen on all addresses)	Local IP address to listen on
<code>SmppListenPort</code>	Integer	2775	Local TCP port to listen on
<code>SmppInactivityTimeout</code>	Long	120000	Time (in ms) before session inactivity timer fires — session has been idle
<code>SmppSessionInitTimeout</code>	Long	30000	Time (in ms) before session init timer fires — session was not bound in time after TCP connection opened
<code>SmppEnquireLinkTimeout</code>	Long	60000	Time (in ms) before enquire link timer fires — app should send an ENQUIRE_LINK PDU
<code>SmppResponseReceivedTimeout</code>	Long	10000	Time (in ms) that stack waits for a response to be received
<code>SmppResponseSentTimeout</code>	Long	10000	Time (in ms) that stack waits for a response to be sent (to an incoming request)
<code>SmppStateCheckingEnabled</code>	Boolean	true	Determines whether stack checks for correct session state when sending/receiving messages.

SMPP Bound RA configuration

Property	Type	Default	Description
<code>SMSCHost</code>	String	localhost	IP address or hostname of SMSC that this entity will connect to.
<code>SMSCPort</code>	Integer	2775	TCP port to connect to.
<code>BindMode</code>	String	TRX	Type of bind request to use — must be one of TX, RX, or TRX (transmitter, receiver or transceiver).
<code>SystemID</code>	String	rhino	SystemID in bind request for authenticating to SMSC.

Password	String	rhino	Password in bind request for authenticating to SMSC.
RebindInterval	Long	5000	Interval (in ms) between re-bind attempts after connection dies.
EnquireLinkInterval	Long	5000	Interval (in ms) between ENQUIRE_LINK requests. The RA periodically sends these to check the session is still active.
EnquireLinkThreshold	Integer	5	Number of consecutive failed ENQUIRE_LINK requests, after which the RA will drop the connection and attempt to reconnect to the SMSC.
SmppResponseReceivedTimeout	Long	10000	Time (in ms) that stack waits for a response to be received.
SmppResponseSentTimeout	Long	10000	Time (in ms) that stack waits for a response to be sent (to an incoming request). The RA will automatically send a GENERIC_NACK if the SBB has not sent a response.

2 Running the example SMPP RA services

Simple MC, and Routing services

The SMPP package contains three example services: [Simple](#), [Message Centre](#), and [Routing](#). Each has a target in the build script to deploy the example with all its dependencies.

- ✔ Use `ant -p` to see the available targets.
- ℹ The source code for the examples is in `examples/src`.

SMPP Simple example

This SBB receives `DELIVER_SM` PDUs and sends responses. It uses the **SMPP Bound** RA.

- 1 Run the `deploy-simple` target in the supplied Ant build file:

```
ant deploy-simple
```

- 2 Run the `smsc_deliver_sm.sh` script:

```
bin/smsc_deliver_sm.sh -t -v -p 2775 -a 012345 -r 1
```

SMPP MC example

This SBB is a very simple MC/SMSC that does nothing other than respond to `bind`, `unbind` and `submit_sm` SMPP commands.

It uses the **SMPP Generic** RA.

- 1 Run the `deploy-smpp-mc` target in the supplied Ant build file:

```
ant deploy-smpp-mc
```

- 2 Run the `smpp_test_client.sh` script:

```
bin/smpp_test_client.sh -h localhost -v
```

- 3 Use the menu options to open a session (3), send `bind` (5), `submit_sm` (12), `unbind` (11), and close the session (4).

- 4 ⚠ The SMPP Generic resource adaptor must be undeployed (or at least deactivated) before running the other examples, because it uses the same port:

```
ant undeploy-smpp-mc  
ant undeploy-smpp-ra
```


SMPP Routing example

This service uses three instances of the **SMPP Bound** RA. When it receives a `DELIVER_SM` event, it responds with a `DELIVER_SM_RESP` message, and then sends a `SUBMIT_SM` request to an SMSC, using the routing table. This mimics the operation of a messaging application where an acknowledgement short message is sent back to the mobile subscriber.

- 1 Run the `deploy-router-client` target in the supplied Ant build file:

```
ant deploy-router-client
```

- 2 Start three instances of the `smsc_deliver_sm.sh` tool, with ports 2775, 12775, and 12776.

```
bin/smsc_deliver_sm.sh -t -v -p 2775 -a 0000 -r 1  
bin/smsc_deliver_sm.sh -t -v -p 12775 -a 0001 -r 1  
bin/smsc_deliver_sm.sh -t -v -p 12776 -a 0002 -r 1
```

- 3 Use the port 2775 instance to send the requests.
The other two instances will receive and respond to `SUBMIT_SM` messages.