EDISON Software Development Centre

SMPP-gateway software manual

SMS2Serve Router Component

http://www.sms2serve.com

EDISON
01.10.2010

Table of Contents

Introduction

SMS2Serve Router – is a reliable high-performance SMS-router that is easy to set up. Its great performance makes it possible to overcome many of the difficulties that occur within the working process. For instance, the presence of damaged network components, network attacks, restarts following power failures and extremely high user traffic. The gateway is based on SMPP v3.4 protocol (Issue 1.2, 11/12/1999).

This manual provides introductory information on system features and configuration.

Features

Our reliable tried and tested telecommunications platform provides the following features:

1. Compatibility with different types of OS: Windows, Linux, Solaris and MacOS
2. SMPP, XML, HTTP, SMTP protocol support
3. Available platforms - Java, .NET
4. Message routing
5. Separate message queue for each connector
6. Convenient VSME-module development
7. High system performance designed to enable the processing of 30 sms messages per second

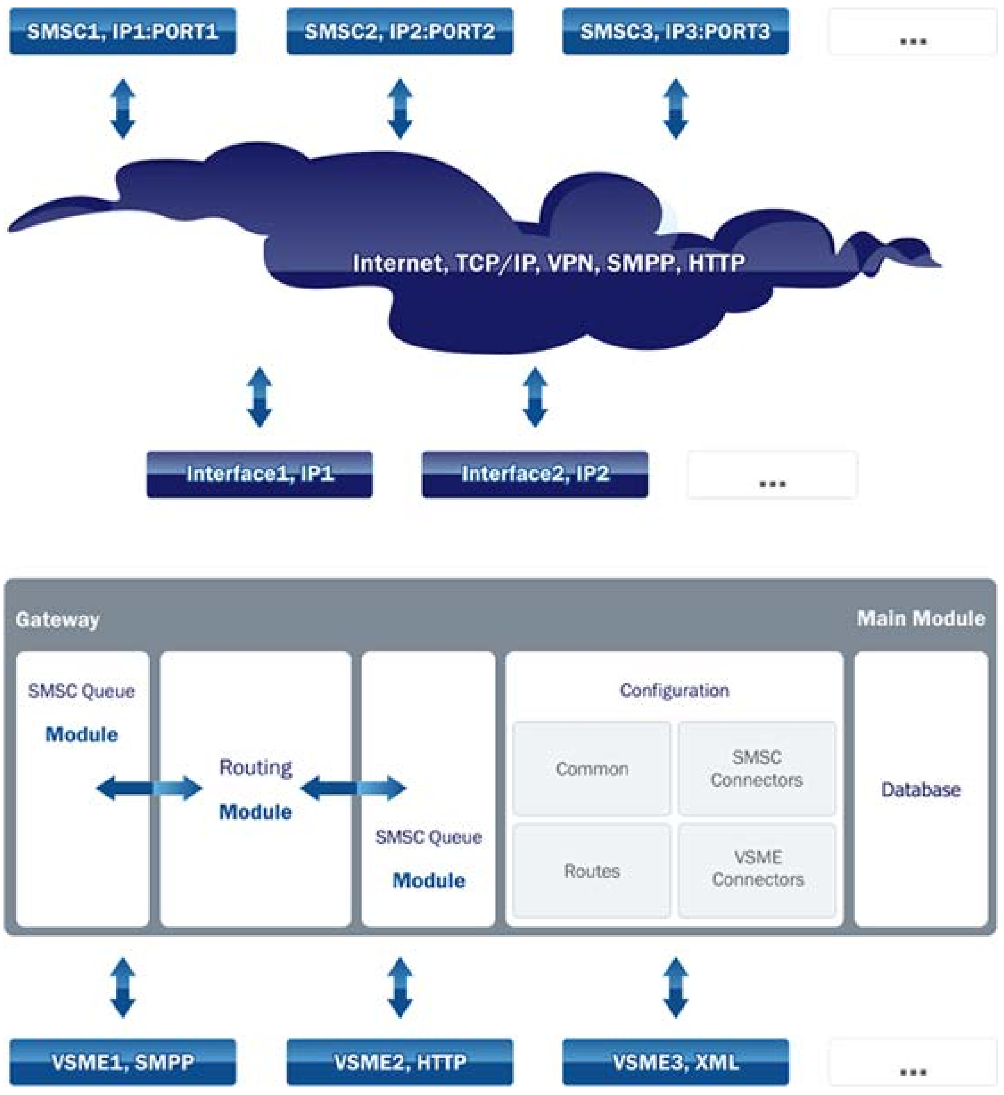
Functions

The main function of the gateway is to “virtualize” the connection to the Short Message Servicing Centre (SMSC) and to support the several virtual SMS-services (VSME) that work with it.

Main components:

1. SMSC queue module
2. Routing module
3. VSME queue module
4. Configuration
5. Database

A schematic model of the gateway is shown in this diagram:



SMSC queue module

The gateway establishes a connection with the SMSC using various IP addresses. One primary IP address is selected with the others acting as backups. If there is a loss of connection on the main interface, a new connection is established via the backup interfaces. The connection status can be checked using the

ENQUIRE_LINK command that is sent to all SMSCs. There are different types of SMSC connectors for different SMSCs. There is also a timer for every SMSC connection.

Supported modes include - transmitter, transceiver, receiver, synchronous and non-synchronous modes (for every SMSC).

The SMSC queue module only transfers messages to the routing module if they are related to SMS transfer.

Routing module

The routing module provides routing and bidirectional message transmission between the SMSC and VSME. Path selection is based on route configuration data analysis.

Routing rules for the text contained in messages allow for the following templates:

(*) – random quantity of any symbols

(??) – any symbol

(?0) – any number

(?a) – any letter

The rules are outlined in the form of conditions and are similar to those of program firewalls. If a transmission route is not specified, an appropriate line will be allocated (this may be configured in setting).

VSME queue module

The connection between the VSME and the gateway is provided through the TCP/IP. The connection between the VSME and the gateway is verified using connection data (login and password). If the connection is lost, the system will reestablish it without informing the VSME. All incoming messages received at the time the VSME connection was lost are saved to the database and are sent to the VSME immediately after the connection is reestablished.

Database

MySQL and MS SQL are used as a database. Delayed messages are stored in a certain table within the database.

Statistics

The gateway supports the gathering of statistics on incoming and outgoing SMS messages within the database. The table of statistics includes the following fields:

Datetime – date and time the message was sent

Direction – input or output

Fconnector – name, incoming connector index

Tconnector – name, outgoing connector index

Addr – address to which the message is sent, or the one that sent the message

Abonent – subscriber's phone number

Message – text of the message

Log

The system log records events and data relating to them, including: event time, module, event type and detailed information on it. There are several logging regimes - from fatal error logging to test mode. While using test mode the system records all possible events.

System Requirements

Work using the Linux operating system on the server (one processor Intel 2 GHz, 256 Mb RAM). It is possible to work with several processors and system tweaks on the Sun Solaris platform.

SMPP Router Configuration

Various SMPP Router settings may be adjusted in the configuration file: smpp_router_debug.conf. This file is a text document that includes a list of settings and their values, in the following format:

```
Parameter = value
```

If there are several values, they are divided by the symbol ";" (without quotation marks). In order to transfer a new value to a new line, the symbol "\" (without quotation marks) must be used at the end of the previous line. The list of values must end with the symbol ";" (without quotation marks).

Example:

```
Parameter = value1; value2;\
                    value3;
```

For the sake of convenience, the contents of the configuration file are divided into the following sections:

1. Program control settings
2. Logging module settings

3. Database connection settings
4. Report module settings
5. SMSC connectors default settings
6. VSME connectors default settings
7. SMSC connector settings (this section is described for every connector)
8. VSME connector settings (this section is described for every connector)

Below is a detailed description of each parameter and its value by section.

Program control settings:

control.port	local TCP/IP port number, which is responsible for program control (e.g. STOP and RESTART commands, etc).
control.encoding	control.encoding encoding of text messages transferred between the basic program and control program. Admissible settings values: Cp866, Cp1251, Windows-1251, KOI8_R, ISO8859_5, MacCyrillic

Logging module settings

logger.file.level, **logger.console.level**	Logging level. Only messages that have an equal or higher level than that specified are recorded. The logging level can be specified either by a text value or a digital value in decimal format. The table of values is shown below:

Code	**Entry**	**Description**
MaxInt	OFF	Logging disabled.
1100	FATAL	Normal functioning is impossible after this error has occurred.
1010	EXCEPTION	Unexpected program error.
1001	ERROR	Program execution error.
0900	WARNING	Warning message.
0802	PDU_TRACE	Command acceptance confirmation message
0800	INFO	Notification..
0700	CONFIG	Program configuration information
0303	DEBUG	Program execution tracing.
MinInt	ALL	Logging of all messages.

logger.file.encoding, **logger.console.encoding**	Log file code list Admissible settings values: Cp866, Cp1251, Windows-1251, KOI8_R, ISO8859_5, MacCyrillic
logger.file.dir	Directory, where log files are stored. A relative path.

Database connection settings

database.driver_packet	package file name with drivers for database connection.
database.driver_class	name of driver type or class for database connection.

database.connection_str	database connection entry.
database.user_name	database username.
database.user_pwd	database username password.
database.time_live_router_info	information storage time (measured in minutes) in the routing table.

Report module settings

report.class	Class responsible for the sending of reports Possible values: *ru.edsd.report.ReportMail* - sends reports via e-mail.
report.enabled	switch that allows a report to be sent. Possible values: true – switch is on, false – switch is off
Settings for ru.edsd.report.ReportMail:	
report.mail.host	host, where mail server is located.
report.mail.from	e-mail address from which the report was sent.
report.mail.to	e-mail address to which the report is sent.
report.mail.subject	subject of the sent letter with a report.

SMSC connectors default settings

SMSC.log_level	logging level for all SMSC connectors.

Code	**Entry**	**Description**
MaxInt	OFF	Logging disabled.
1100	FATAL	Normal functioning is impossible after this error has occurred.
1010	EXCEPTIO N	Unexpected program error.
1001	ERROR	Program execution error.
0900	WARNING	Warning message.
0802	PDU_TRAC E	Command acceptance confirmation message.
0800	INFO	Notification.
0700	CONFIG	Program configuration information.
0303	DEBUG	Program execution tracing.
MinInt	ALL	Logging of all messages.

The value of this parameter cannot be lower than the value of global logging parameter.

SMSC.timeout_shutdown	time period (measured in seconds) for expected session shutdown. If a session does not finish in the time specified and in the usual way, it will be terminated automatically.
SMSC.enquire_link.send	Switch that gives permission to send an ENQUIRE_LINK request if there is no similar request from the remote host during the time period specified within the parameter *SMSC.enquire_link.timeout_wait*. Possible values: true – switch is on, false – switch is off
SMSC.enquire_link.timeout_wait_resp	period of time (measured in seconds) spent waiting for a response to the ENQUIRE_LINK command. This is only used when:

	SMSC.enquire_link.send = true. If there is no reply within a specified time period, the session will be terminated.
SMSC.enquire_link.timeout_wait	period of time (measured in seconds) spent waiting for a response to the ENQUIRE_LINK command from the remote host. If there is no response within a specified time period then depending on the *SMSC.enquire_link.send* parameter, the following operations will be carried out: if *SMSC.enquire_link.send = true* the system will send its own ENQUIRE_LINK request if *SMSC.enquire_link.send = false* the system will terminate the session because it will believes the remote host is disconnected.
SMSC.protocol.timeout_err_net	time period (measured in seconds) for the delay in network operations if a previous network operation with a socket failed.
SMSC.protocol.timeout_stop_send	time period (measured in seconds) for delays in message-dispatch if a *_RESP command with a status code defined in *SMSC.protocol.status_stop_send* is received in response to one of the commands.
SMSC.protocol.timeout_wait_unbind_resp	period of time (measured in seconds) spent waiting for a response to a UNBIND command. If =0, there will be no time spent waiting for a response.
SMSC.protocol.status_stop_send	status code list (see appendix 1 - list of status codes for which message-dispatch delay is provided within the time period specified in *SMSC.protocol.timeout_stop_send*).
SMSC.connect.reconnect	switch indicating the need to reconnect if a previous connection attempt was unsuccessful or the connection with a remote host was lost.
SMSC.connect.timeout_error_report	time period (measured in seconds) immediately after which the system sends a recurrent report indicating a connection problem with a remote host.
SMSC.connect.try_count	number of connection attempts to a remote host. If it is not possible to establish a connection within a specified number of attempts, an appropriate report is created. If the *SMSC.connect.reconnect* switch is off, then there will not be any attempts to reconnect.
SMSC.connect.try_delay	time period (measured in seconds) that indicates the delay between connection attempts (see *SMSC.connect.try_count*).
SMSC.connect.time_out	period of time (measured in seconds) spent waiting for a connection at the physical layer.
SMSC.connect.timeout_wait_bind	period of time (measured in seconds) spent waiting for a response to a BIND command. If there is no response within a specified time, the connection is considered unsuccessful, the physical connection is terminated and an attempt to reconnect is made.
SMSC.connect.from_host	list of IP addresses from which connection attempts are made to a remote host. The first address on the list is taken to be the primary address and the others act as backup addresses.
SMSC.route.status_not_route	a status code used to respond to the SMSC if it is impossible to define the route of a specified message.
SMSC.route.text_sms_not_route	SMS text that is sent to the SMSC if it is impossible to define a route and if *SMSC.route.status_not_route = ESME_ROK (0).*
SMSC.route.status_not_connect	a status code used to respond to the SMSC if there is no connection at the VSME connector.
SMSC.route.text_sms_not_connect	SMS text that is sent to the SMSC if there is no connection at the VSME connector or if *SMSC.route.status_not_connect = ESME_ROK (0).*

SMSC.route.mask	gives a list of routes in the following formats: *source_connector, dest_connector, msg_mask, dest_addr_mask;* Where source_connector is a source connector name (only SMSC), dest_connector is a receiver connector name (only VSME), msg_mask – short message mask (it is possible to use regular expressions), dest_addr_mask – destination address mask (it is possible to use regular expressions)
SMSC.route.status_not_connect_database	a status code used to respond to the SMSC if there is no database connection.

VSME connectors default settings

VSME.log_level	logging level for all VSME connectors.

Code	**Entry**	**Description**
MaxInt	OFF	Logging disabled.
1100	FATAL	Normal functioning is impossible after this error has occurred.
1010	EXCEPTION	Unexpected program error.
1001	ERROR	Program execution error.
0900	WARNING	Warning message.
0802	PDU_TRACE	Command acceptance confirmation message.
0800	INFO	Notification.
0700	CONFIG	Program configuration information.
0303	DEBUG	Program execution tracing.
MinInt	ALL	Logging of all messages.

	The value of this parameter cannot be lower than the value of global logging parameter.
VSME.timeout_shutdown	time period (measured in seconds) for expected session shutdown. If a session does not finish in the time specified, it will be terminated automatically.
VSME.enquire_link.send	Switch that gives permission to send an ENQUIRE_LINK request if there is no similar request from the remote host during the time period specified within the parameter *VSME.enquire_link.timeout_wait.*
VSME.enquire_link.timeout_wait_resp	time period (measured in seconds) spent waiting for a response to an ENQUIRE_LINK command. This is only used when *VSME.enquire_link.send = true*. If there is no reply within a specified time period, the session will be terminated.
VSME.enquire_link.timeout_wait	period of time (measured in seconds) spent waiting for a ENQUIRE_LINK command request from the remote host. If no request arrives within a specified time period then depending on the *VSME.enquire_link.send* parameter, the following operations will be carried out: if *VSME.enquire_link.send = true* the system will send its own ENQUIRE_LINK request if *VSME.enquire_link.send = false* the system will terminate the session because it believes the remote host to be disconnected.
VSME.protocol.timeout_err_net	time period (measured in seconds) for the delay in network operations if a previous network operation with a socket failed.

VSME.protocol.timeout_stop_send	time period (measured in seconds) for delays in message-dispatch if a *_RESP command with a status code defined in *VSME.protocol.status_stop_send* is received in response to one of the commands.
VSME.protocol.status_stop_send	list of status codes for message-dispatch delay for the time period indicated in *VSME.protocol.timeout_stop_send.*
VSME.protocol.timeout_wait_unbind_resp	period of time (measured in seconds) spent waiting for a response to a UNBIND command. If =0, there will be no time spent waiting for a response.
VSME.connect.reconnect	switch indicating the need to reconnect if a previous connection attempt was unsuccessful or the connection with a remote host was lost.
VSME.connect.timeout_error_report	time period (measured in seconds) immediately after which the system sends a recurrent report indicating a connection problem with a remote host.
VSME.connect.try_count	number of connection attempts to a remote host. If it is not possible to establish a connection within a specified number of attempts, an appropriate report is created. If the *VSME.connect.reconnect* switch is off, then there will not be any attempts to reconnect.
VSME.connect.try_delay	time period (measured in seconds) that indicates the delay between connection attempts (see *VSME.connect.try_count*).
VSME.connect.time_out	period of time (measured in seconds) spent waiting for a connection at the physical layer.
VSME.connect.timeout_wait_bind	period of time (measured in seconds) spent waiting for a response to a BIND command. If there is no response within a specified time, the connection is considered unsuccessful, the physical connection is terminated and an attempt to reconnect is made.
VSME.connect.accept_host	list of IP addresses used for connection attempts to a remote host (i.e. the TCP/IP port is set to open or “accept”). The first address on the list is the primary address and the others act a backup addresses.
VSME.script.dir	directory, where script files are stored
VSME.route.status_not_route	a status code used to respond to the VSME if it is impossible to define the route of a given message.
VSME.route.status_not_connect	a status code used to respond to the VSME if there is no connection at the SMSC connector.
VSME.route.status_not_connect_database	a status code used to respond to the VSME if there is no database connection.

SMSC connector settings (this section is described for every connector)

It is possible to set any value for parameters from the SMSC default section. However, the default value for this given connector will be cancelled.

{имя коннектора}	the connector ID must have a “connector” value. If this value is not specified or is incorrect, this connector will not be initialized.
{имя коннектора}.class	connector class name Possible values: *ru.edsd.smpp_router.connector.SMSCAsyncConnector* – connector used for SMSC connection. This connector can provide a non-synchronic algorithm for message-dispatch queues. *ru.edsd.smpp_router.connector.SMSCSyncConnector* - connector

	used for SMSC connection. This connector can provide a synchronic algorithm for message-dispatch queues.
{имя коннектора}.connect.host	name of the remote host, where the SMSC is installed.
{имя коннектора}.connect.port	TCP/IP port used by the SMSC
{имя коннектора}.bind.type	connection type, which defines the types of interaction with the SMSC. Possible values: BIND_RECEIVER BIND_TRANSMITTER BIND_TRANSCEIVER
{имя коннектора}.bind.system_id	BIND command settings, which are used to connect to the SMSC.
{имя коннектора}.bind.password	authentication password.
{имя коннектора}.bind.system_type	optional parameter used for VSME category indication.
{имя коннектора}.bind.addr_ton	indicates VSME address type. Possible values: 00 - UNKNOWN, 01 - INTERNATIONAL, 02 - NATIONAL, 03 - NETWORK, 04 - SUBSCRIBER, 05 - ALPHANUMERIC, 06 – ABBREVIATED. It is possible to set both text and decimal values.
{имя коннектора}.bind.addr_npi	shows the VSME number plan indicator. Possible values: 00 - UNKNOWN, 01 - ISDN, 03 - DATA, 04 - TELEX, 06 - LAND MOBILE, 08 - NATIONAL, 09 - PRIVATE, 10 - ERMES, 14 - INTERNET, 18 - WAP CLIENT ID. It is possible to set both text and decimal values.
{имя коннектора}.bind.address_range	address or range of addresses served by the VSME.

Секция параметров для коннектора VSME (данная секция описывается для каждого коннектора

{имя коннектора}	the connector ID must have a “connector” value. If this value is not specified or is incorrect, this connector will not be initialized.
{имя коннектора}.class	connector class name Possible values: *ru.edsd.smpp_router.connector.VSMEAsyncConnector* – connector used for VSME connection. This connector can provide a non-synchronic algorithm for message-dispatch queues. *ru.edsd.smpp_router.connector.VSMESyncConnector* – connector used for VSME connection. This connector can provide a synchronic algorithm for message-dispatch queues.
{имя коннектора}.connect.accept_port	TCP/IP port where the connection with be established
{имя коннектора}.bind.system_id	BIND command settings, which will be used to indicate a successful connection. Invalid settings will cause connection request denial and session termination.
{имя коннектора}.bind.password	authentication password.
{имя коннектора}.script.start	script that will be used before initializing a physical connection. The script must be located in the “script” folder. However, only the file name is specified in the settings (without subfolder indication). If there is no value for this parameter then no action will be carried out.
{имя коннектора}.script.stop	script that will be used after the termination of a physical connection or in case of a connection error.

Appendix 1 – List of status codes

0x00000000	ESME_ROK	No error
0x00000001	ESME_RINVMSGLEN	Message Length is invalid
0x00000002	ESME_RINVCMDLEN	Command Length is invalid
0x00000003	ESME_RINVCMDID	Invalid Command ID
0x00000004	ESME_RINVBNDSTS	Incorrect BIND Status for given command
0x00000005	ESME_RALYBND	ESME Already in bound state
0x00000006	ESME_RINVPRTFLG	Invalid priority flag
0x00000007	ESME_RINVREGDLVFLG	Invalid registered delivery flag
0x00000008	ESME_RSYSERR	System Error
0x0000000A	ESME_RINVSRCADR	Invalid source address
0x0000000B	ESME_RINVDSTADR	Invalid destination address
0x0000000C	ESME_RINVMSGID	Message ID is invalid
0x0000000D	ESME_RBINDFAIL	Bind failed
0x0000000E	ESME_RINVPASWD	Invalid password
0x0000000F	ESME_RINVSYSID	Invalid System ID
0x00000011	ESME_RCANCELFAIL	Cancel SM Failed
0x00000013	ESME_RREPLACEFAIL	Replace SM Failed
0x00000014	ESME_RMSGQFUL	Message queue full
0x00000015	ESME_RINVSERTYP	Invalid service type
0x00000033	ESME_RINVNUMDESTS	Invalid number of destinations
0x00000034	ESME_RINVDLNAME	Invalid distribution list name
0x00000040	ESME_RINVDESTFLAG	Destination flag is invalid (submit_multi)
0x00000042	ESME_RINVSUBREP	Invalid `submit with replace' request (i.e. submit_sm with replace_if_present_flag set)
0x00000043	ESME_RINVESMCLASS	Invalid esm_class field data
0x00000044	ESME_RCNTSUBDL	Cannot submit to distribution list
0x00000045	ESME_RSUBMITFAIL	submit_sm or submit_multi failed
0x00000048	ESME_RINVSRCTON	Invalid source address TON
0x00000049	ESME_RINVSRCNPI	Invalid source address NPI
0x00000050	ESME_RINVDSTTON	Invalid destination address TON
0x00000051	ESME_RINVDSTNPI	Invalid destination address NPI
0x00000053	ESME_RINVSYSTYP	Invalid system_type field
0x00000054	ESME_RINVREPFLAG	Invalid replace_if_present flag
0x00000055	ESME_RINVNUMMSGS	Invalid number of messages
0x00000058	ESME_RTHROTTLED	Throttling error (ESME has exceeded allowed message limits)
0x00000061	ESME_RINVSCHED	Invalid scheduled delivery time
0x00000062	ESME_RINVEXPIRY	Invalid message validity period (expiry time)
0x00000063	ESME_RINVDFTMSGID	Predefined message invalid or not found
0x00000064	ESME_RX_T_APPN	ESME Receiver Temporary App Error Code
0x00000065	ESME_RX_P_APPN	ESME Receiver Permanent App Error Code
0x00000066	ESME_RX_R_APPN	ESME Receiver Reject Message Error Code
0x00000067	ESME_RQUERYFAIL	query_sm request failed
0x000000C0	ESME_RINVOPTPARSTREAM	Error in the optional part of the PDU Body.
0x000000C1	ESME_ROPTPARNOTALLWD	Optional Parameter not allowed
0x000000C2	ESME_RINVPARLEN	Invalid Parameter Length.
0x000000C3	ESME_RMISSINGOPTPARAM	Expected optional parameter missing
0x000000C4	ESME_RINVOPTPARAMVAL	Invalid optional parameter value

0x000000FE	ESME_RDELIVERYFAILURE	Delivery Failure (used for data_sm_resp)
0x000000FF	ESME_RUNKNOWNERR	Unknown error

Appendix 2 – Demonstration configuration file

```
############################################################################

control.port          = 3500

control.encoding        = KOI8_R

############################################################################

logger.file.dir         = logs

logger.file.level        = INFO

logger.file.encoding     = Cp1251

logger.console.level     = INFO

logger.console.encoding  = Cp1251


############################################################################

database.driver_packet        = mysql-connector-java-3.0.16-ga-bin.jar

database.driver_class         = com.mysql.jdbc.Driver

database.connection_str                                                        =
jdbc:mysql://127.0.0.1/smpp_router?useUnicode=true&characterEncoding=Cp1251

database.user_name            = root

database.user_pwd             = 123

database.time_live_router_info = 60

############################################################################

report.enabled          = false

report.mail.host         =

report.mail.from         =

report.mail.to          =
report.mail.subject      = SMPProuter ALERT

#############################################################################

SMSC.log_level                     = ALL

SMSC.timeout_shutdown                = 10

SMSC.enquire_link.send               = true

SMSC.enquire_link.timeout_wait        = 60

SMSC.enquire_link.timeout_wait_resp    = 3

SMSC.protocol.timeout_err_net         = 60

SMSC.protocol.timeout_stop_send        = 30

SMSC.protocol.timeout_wait_unbind_resp = 5

SMSC.protocol.status_stop_send        = ESME_RMSGQFUL;\

                                    ESME_RTHROTTLED;

SMSC.connect.reconnect               = true

SMSC.connect.timeout_error_report      = 1800

SMSC.connect.try_count               = 2

SMSC.connect.try_delay               = 30

SMSC.connect.time_out                = 20

SMSC.connect.timeout_wait_bind         = 10

SMSC.connect.from_host               = localhost;

SMSC.route.status_not_route           = ESME_ROK

SMSC.route.text_sms_not_route          = Cannot define VSME

SMSC.route.status_not_connect          = ESME_ROK

SMSC.route.text_sms_not_connect         = No connect VSME

SMSC.route.status_not_connect_database = ESME_RSYSERR

#############################################################################

VSME.log_level                     = ALL
VSME.timeout_shutdown                  = 10
VSME.script.dir                        = script
VSME.enquire_link.send                 = true
VSME.enquire_link.timeout_wait         = 30
VSME.enquire_link.timeout_wait_resp    = 2
VSME.protocol.timeout_err_net          = 30
VSME.protocol.timeout_stop_send        = 30
VSME.protocol.timeout_wait_unbind_resp = 5
VSME.protocol.status_stop_send         = ESME_RMSGQFUL;\
                                         ESME_RTHROTTLED;
VSME.connect.reconnect                 = true
VSME.connect.timeout_error_report      = 1800
VSME.connect.try_count                 = 3
VSME.connect.try_delay                 = 1
VSME.connect.time_out                  = 60
VSME.connect.timeout_wait_bind         = 10
VSME.connect.accept_host               = localhost;
VSME.route.status_not_route            = ESME_RINVDSTADR
VSME.route.status_not_connect          = ESME_RTHROTTLED
VSME.route.status_not_connect_database = ESME_RSYSERR
##############################################################################
SMSC_test_1                            = connector
SMSC_test_1.class                      = ru.imtco.smpp_router.connector.SMSCSyncConnector
SMSC_test_1.connect.host               = localhost
SMSC_test_1.connect.port               = 3100
SMSC_test_1.bind.type                  = BIND_TRANSCEIVER
```

```
SMSC_test_1.bind.system_id          = test_1
SMSC_test_1.bind.password           = test_1
SMSC_test_1.bind.system_type         = logic
SMSC_test_1.bind.addr_ton           = UNKNOWN
SMSC_test_1.bind.addr_npi           = UNKNOWN
SMSC_test_1.bind.address_range        =
#############################################################################
VSME_test_1                         = connector
VSME_test_1.class                     = ru.imtco.smpp_router.connector.VSMEAsyncConnector
VSME_test_1.connect.accept_port          = 4100
VSME_test_1.bind.system_id              = test
VSME_test_1.bind.password               = 123
VSME_test_1.script.start               = start_vsme_1.bat
VSME_test_1.script.stop                =
#############################################################################
VSME_test_2                         = connector
VSME_test_2.class      =  ru.imtco.smpp_router.connector.VSMEAsyncConnector
VSME_test_2.connect.accept_port          = 4200
VSME_test_2.bind.system_id              = test
VSME_test_2.bind.password               = 123
VSME_test_2.script.start               =
VSME_test_2.script.stop                =
#############################################################################
SMSC.route.mask                    = SMSC_test_1,  VSME_test_1,   TEST1 .*,   (.|\n)*;\
                                     SMSC_test_1,  VSME_test_1,   BURDA?,     (.|\n)*;\
                                     SMSC_test_1,  VSME_test_2,   TEST2 .*,   (.|\n)*;
```

##