

# External Short Messaging Entities (ESMEs)

# Table of Contents

Create new ESME .....	3
Using CLI .....	3
Using GUI .....	10
Modify ESME .....	11
Using CLI .....	11
Using GUI .....	17
View ESME Details .....	18
Using CLI .....	18
Using GUI .....	18
Delete an existing ESME .....	19
Using CLI .....	19
Using GUI .....	19
Start ESME .....	19
Using CLI .....	19
Using GUI .....	20
Stop ESME .....	20
Using CLI .....	20
Using GUI .....	20
Other ESME Operations .....	21
Using GUI .....	21

&THIS.PLATFORM;&THIS.APPLICATION; can now act as ESME (initiate bind to remote SMSC) or can also act as SMSC (accept bind from remote ESME). While defining an ESME (SMPP connection), you can optionally pass the Cluster name. If it is not passed, cluster name is same as ESME name. As the name suggests, it is now possible to group different ESMEs in the same cluster. This is useful only when SMS is supposed to be routed out of &THIS.PLATFORM;&THIS.APPLICATION; to ESME. If there are multiple ESMEs in a cluster, the load is shared in a round robin fashion to send out SMS. In case if one of the ESME is in **UNBOUND** state, the next **BOUND** smpp connection (within same cluster) will be used. Below diagram explains the load-balancing between SMPP connections

### Case 1 : SMSC acting as CLIENT



### Case 2 : SMSC acting as SERVER



### Case 3 : SMSC acting as SERVER - Load balancing via HAProxy

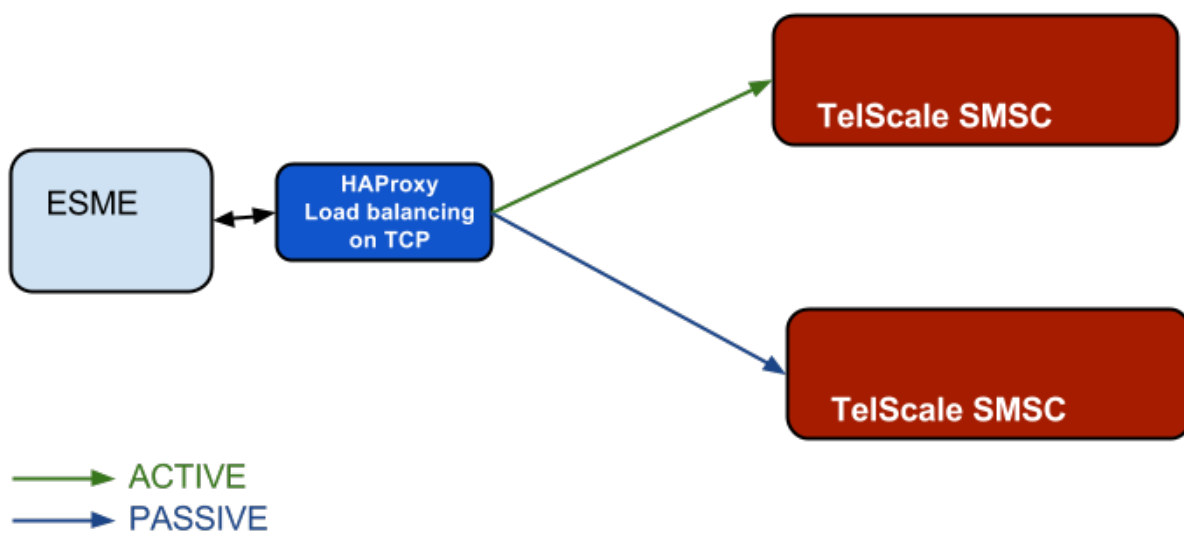


Figure 1. SMPP Load Balancing

You can define multiple ESMEs, each with a unique name but the same **systemId** and declare their **host-ip** and **port** values as -1 (only for SMPP type Server). The SMSC will now accept multiple incoming BIND requests from any **IP** and any **port** as far as the **systemId** and **Password** matches.

Alternatively, if you declare a real value for **host-ip**(say for example 10.199.7.23) and **port** as -1, the SMSC will accept as many BINDS as ESMEs defined from the specified IP but any port as far as the **systemId** and **Password** matches.

## Create new ESME

### Using CLI

You can configure a new ESME by issuing the command ``smpp esme create`` with appropriate parameters as described below.

Name

```
smpp esme create
```

#### SYNOPSIS

```
smpp esme create name <systemId> <host-ip> <port> <SmppBindType>
<SmppSession.Type> password <password> networkid <networkId>
system-type <sms | vms | ota >
interface-version <3.3 | 3.4 | 5.0> esme-ton <esme address ton>
esme-npi <esme address npi> esme-range <esme address range>
cluster-name <clusterName> window-size <windowSize>
connect-timeout <connectTimeout> request-expiry-timeout <requestExpiryTimeout>
window-monitor-interval <windowMonitorInterval>
window-wait-timeout <windowWaitTimeout> counters-enabled <true | false>
enquire-link-delay <30000> charging-enabled <true | false>
source-ton <source address ton> source-npi <source address npi>
source-range <source address range> routing-ton <routing address ton>
routing-npi <routing address npi> routing-range <routing address range>
ratelimit-second <ratelimitsecond> ratelimit-minute <ratelimitminute>
ratelimit-hour <ratelimithour> ratelimit-day <ratelimitday>
min-message-length <min-message-length value>
max-message-length <max-message-length value>
national-language-locking-shift <NationalLanguageIdentifier value>
national-language-single-shift <NationalLanguageIdentifier value>
```

#### DESCRIPTION

This command is used to configure a new ESME.

#### PARAMETERS

## Standard Parameters

- Name** - A unique name for this ESME configuration. You can define as many ESMEs as you want as far as the name is unique and the combination of SystemId:host-ip:port:SmppBindType is unique.
- System Id** - This is used to identify an ESME or an SMSC at bind time. An 'ESME system\_id' identifies the ESME or ESME agent to the SMSC. The 'SMSC system\_id' provides an identification of the SMSC to the ESME. You can define multiple ESMEs, each with a unique name but the same <literal>systemId</literal> to allow anonymous incoming binds and multiple binds from the same IP depending on the values specified for host-ip and port.
- host-ip & port** - If the SMSC is acting as an ESME, the BIND request will be sent to the configured IP and Port. If the SMSC is acting as a Server, it will accept incoming BIND requests from the specified IP and Port. If the port is unknown, you must pass '-1' as wild character.

When you define multiple ESMEs with the same systemId, if host-ip and port values are -1 (for SMPP type Server), the SMSC will accept multiple incoming BIND requests from any IP:port as long as the systemId and password match.

When you define multiple ESMEs with the same systemId, if host-ip is a real value (a specific IP) and port value is -1, the SMSC will accept as many BINDS as ESMEs defined from the specified IP but any port as long as the systemId and password match.

- SmppBindType** - Possible values: TRANSCEIVER, TRANSMITTER or RECEIVER. If the SMSC is acting as an ESME, it will initiate corresponding bind. If the SMSC is acting as a Server, it will accept corresponding bind from a remote ESME.
- SmppSession.Type** - Possible values: SERVER or CLIENT. If the value is 'SERVER', the SMSC acts as a Server listening for incoming SMPP binds. If the value is 'CLIENT', the SMSC will initiate SMPP bind to a remote Server.

## Optional Parameters

- Password** - It is used by the SMSC to authenticate the identity of the binding ESME. The Service Provider may require ESME's to provide a password when binding to the SMSC.

networkId	<p>- indicates virtual subnetwork that this ESME belongs. SMS flows within same networkId, unless changed using mproc (this is for multi-tenancy support). If this parameter is skipped - networkId will be set to "0" when ESME creation. If you do not use multi-tenancy support - set this value to 0 or skip.</p>
system-type	<p>- Default value is null. This is used to categorize the type of ESME that is binding to the SMSC.</p>
interface-version	<p>- Default value is 3.4.</p> <p>It is used to indicate the version of the SMPP protocol.</p> <p>It is set in 'SMPPServer Settings'.</p>
esme-ton	<p>- Defines ESME TON. If the SMPP Session Type is CLIENT, this TON will be used in the BIND request. If the SMPP Session Type is SERVER, the incoming BIND request should have the same TON as configured here. If the configured value is null (-1), SMSC will ignore it in both cases.</p>
esme-npi	<p>- Defines ESME NPI. If the SMPP Session Type is CLIENT, this NPI will be used in the BIND request. If the SMPP Session Type is SERVER, the incoming BIND request should have the same NPI as configured here. If the configured value is null (-1), SMSC will ignore it in both cases.</p>
esme-range	<p>- Defines ESME Address Range. If the SMPP Session Type is CLIENT, this Address Range will be used in the BIND request. If the SMPP Session Type is SERVER, the incoming BIND request should have the same Address Range as configured here. If the configured value is null (-1), SMSC will ignore it in both cases.</p>
cluster-name	<p>- If it is not specified then its same as the name. It is possible to group different SMPP connections together by specifying the same cluster-name. All the SMPP connection's that are capable of sending out SMS are candidates for grouping.</p>
window-size	<p>- Default value is 1. The window size is the amount of unacknowledged requests that are permitted to be outstanding/unacknowledged at any given time. If more requests are added, the underlying stack will throw an exception.</p> <p>This value is set only when ESME is defined as Client side. For Server side this value is taken from the</p>

## 'SMPP Server Settings'.

`connect-timeout` - Default value is 10000 milli seconds.

This parameter is used to specify the time within which the connection to a remote SMSC server should be established.

This is useful only when ESME is defined as Client Side. For Server side this value is taken from the the 'SMPP Server Settings'.

`request-expiry-timeout` - Default value is -1 (disabled).

This parameter is used to specify the time to wait in milli seconds for an endpoint to respond to before it expires.

This is useful only when ESME is defined as Client Side. For Server side this value is taken from the the 'SMPP Server Settings'.

`window-monitor-interval` - Default value is -1 (disabled).

This parameter is used to specify the time between executions of monitoring the window for requests that expire. It is recommended that this value, generally, either matches or is half the value of 'request-expiry-timeout'. Therefore, in the worst case scenario, a request could take upto 1.5 times the 'requestExpiryTimeout' to clear out.

This is useful only when ESME is defined as Client Side. For Server side this value is taken from the the 'SMPP Server Settings'.

`window-wait-timeout` - Default value is 60000 milli seconds.

This parameter is used to specify the time to wait until a slot opens up in the 'sendWindow'.

This is useful only when ESME is defined as Client Side. For Server side this value is taken from the the 'SMPP Server Settings'.

`counters-enabled` - Default value is true.

When this is enabled, SMSC exposes the statistics for SMPP connections.

This is useful only when ESME is defined as Client Side. For Server side this value is taken from the the 'SMPP Server Settings'.

`enquire-link-delay` - Default value is 30000 milli seconds.

When SMSC connects to a remote server as CLIENT, it



sends an 'ENQUIRE\_LINK' after every configured enquire-link-delay.

charging-enabled - Flag to enable or disable charging for every SMS arriving from SIP.

source-ton - Every SMS coming into the SMSC via this ESME should have the same 'source\_addr\_ton' as the value configured here.

If this configured value is null(-1) or not null and matches, the SMSC will compare the 'source\_addr\_npi' and 'source\_addr\_range' as explained below.

If it doesn't match, the SMSC will reject this SMS with an error code '0x0000000A' indicating Invalid Source Address.

source-npi - Every SMS coming into the SMSC via this ESME should have the same 'source\_addr\_npi' as the value configured here.

If this configured value is null(-1) or not null and matches, the SMSC will compare the 'source\_addr\_range' as below.

If it doesn't match, the SMSC will reject this SMS with an error code '0x0000000A' indicating Invalid Source Address.

source-range - Every SMS coming into the SMSC via this ESME should have the same 'source\_addr\_range' as the value configured here. This is a regular java expression and default value is `^[0-9a-zA-Z]*`.

If it matches, the SMSC will accept the incoming SMS and process further.

If it doesn't match, the SMSC will reject this SMS with an error code '0x0000000A' indicating Invalid Source Address.

routing-ton - The DefaultSmsRoutingRule will try to match the 'dest\_addr\_ton' of outgoing SMS with the value configured here. If this configured value is null(-1) or not null and matches, the SMSC will compare the 'dest\_addr\_npi' and 'destination\_addr' as explained below. If it doesn't match, the SMSC will select the next ESME in the list for matching routing rule.

DefaultSmsRoutingRule will consider ESME for routing

only if

- 1) SmppBindType is TRANSCEIVER
- 2) SmppBindType is RECEIVER and SmppSession.Type is SERVER
- 3) SmppBindType is TRANSMITTER and SmppSession.Type is CLIENT

routing-npi      - The DefaultSmsRoutingRule will try to match the 'dest\_addr\_npi' of outgoing SMS with the value configured here. If this configured value is null(-1) or not null and matches, the SMSC will compare the 'destination\_addr' as below. If it doesn't match, the SMSC will select the next ESME in the list for matching routing rule.

DefaultSmsRoutingRule will consider ESME for routing only if

- 1) SmppBindType is TRANSCEIVER
- 2) SmppBindType is RECEIVER and SmppSession.Type is SERVER
- 3) SmppBindType is TRANSMITTER and SmppSession.Type is CLIENT

routing-range      - The DefaultSmsRoutingRule will try to match the 'destination\_addr' of outgoing SMS with the value configured here. This is a regular java expression and default value is `^[0-9a-zA-Z]*`. If it matches, the SMSC will send the SMS out over this SMPP connection. If it doesn't match, the SMSC will select the next ESME in the list for matching routing rule.

DefaultSmsRoutingRule will consider ESME for routing only if

- 1) SmppBindType is TRANSCEIVER
- 2) SmppBindType is RECEIVER and SmppSession.Type is SERVER
- 3) SmppBindType is TRANSMITTER and SmppSession.Type is CLIENT

ratelimit\_second      - This parameter is used to specify a maximum limit of messages that the SMSC will accept from this ESME during any one second.

If the ESME sends more messages (per second) than the maximum limit specified by 'ratelimit\_second', these additional messages will be rejected by the SMSC GW along with an error code - "throttled".

The default value for this parameter is "0" and it implies "no restrictions". If this parameter is not specified it implies "no restrictions".

`ratelimit_minute` - This parameter is used to specify a maximum limit of messages that the SMSC will accept from this ESME during any one minute.

If the ESME sends more messages (per minute) than the maximum limit specified by '`ratelimit_minute`', these additional messages will be rejected by the SMSC GW along with an error code - "throttled".

The default value for this parameter is "0" and it implies "no restrictions". If this parameter is not specified it implies "no restrictions".

`ratelimit_hour` - This parameter is used to specify a maximum limit of messages that the SMSC will accept from this ESME during any one hour.

If the ESME sends more messages (per hour) than the maximum limit specified by '`ratelimit_hour`', these additional messages will be rejected by the SMSC GW along with an error code - "throttled".

The default value for this parameter is "0" and it implies "no restrictions". If this parameter is not specified it implies "no restrictions".

`ratelimit_day` - This parameter is used to specify a maximum limit of messages that the SMSC will accept from this ESME during any one day.

If the ESME sends more messages (per day) than the maximum limit specified by '`ratelimit_day`', these additional messages will be rejected by the SMSC GW along with an error code - "throttled".

The default value for this parameter is "0" and it implies "no restrictions". If this parameter is not specified it implies "no restrictions".

`min-message-length` - This parameter is used to specify the minimum message length (in characters) acceptable to the SMSC GW, for messages coming from this ESME.

If an incoming message length is less than the `min-message-length` it will be rejected by SMSC GW.

The default value for this parameter is "-1" and it implies "no limitations". Any other negative value also implies "no limitations".

`max-message-length` - This parameter is used to specify the maximum message length (in characters) acceptable to the SMSC GW, for messages coming from this ESME.

If an incoming message length is more than the `max-message-length` it will be rejected by SMSC GW.

The default value for this parameter is "-1" and it implies "no limitations". Any other negative value also implies "no limitations".

`national-language-locking-shift` - National language locking shift table can be configured for messages that have come via SMPP (this ESME), do not have UDHS inside and have GSM7 encoding (`DCS==0`).

The default GSM data coding table is mostly used.  
Possible values:

- = 0: default GSM data coding table
- = 13: urdu (arabic) national language shift table
- =1: the national language locking shift value must be obtained from the option `national-language-locking-shift` that is defined at SMSC GW general level.

`national-language-single-shift` - National language single shift table can be configured for messages that have come via SMPP (this ESME), do not have UDHS inside and have GSM7 encoding (`DCS==0`).

The default GSM data coding table is mostly used.  
Possible values:

- = 0: default GSM data coding table
- = 13: urdu (arabic) national language single table
- =1: the national language locking shift value must be obtained from the option `national-language-locking-single` that is defined at SMSC GW general level.

## Using GUI

*Procedure: Create new ESME using GUI*

1. In the GUI Management Console for SMSC Gateway, click on 'ESMEs' in the left panel.
2. The main panel will display the existing ESMEs (if any), one each in a row with corresponding actions (start, stop, delete, update) for each row. Below this you will find the button 'Create ESME'.
3. You can create a new ESME by launching the 'Create ESME' window by clicking on the blue coloured 'Create ESME' button. The 'Create ESME' window will display all ESME parameters that must be defined by you. For more details of these parameters please refer to the descriptions of the CLI commands for the same in the preceding section.

4. Enter appropriate values for all the parameters and then click on the 'Create' button at the bottom of this 'Create ESME' window. This action will create a new ESME with parameters as defined by you.
5. If there is an error in defining the ESME, then you will find the details of the error in the Management Console Log section below.

## Modify ESME

### Using CLI

You can modify an existing ESME by issuing the command `smpp esme modify` with appropriate parameters as described below.

#### Name

`smpp esme modify`

#### SYNOPSIS

```
smpp esme modify <name> password <Specify new password>
networkid <networkId>
esme-ton <esme address ton> esme-npi <esme address npi>
esme-range <esme address range> window-size <windowSize>
connect-timeout <connectTimeout> request-expiry-timeout <requestExpiryTimeout>
window-monitor-interval <windowMonitorInterval>
window-wait-timeout <windowWaitTimeout> counters-enabled <true | false>
enquire-link-delay <30000> charging-enabled <true | false>
source-ton <source address ton> source-npi <source address npi>
source-range <source address range> routing-ton <routing address ton>
routing-npi <routing address npi> routing-range <routing address range>
ratelimit-second <ratelimitsecond> ratelimit-minute <ratelimitminute>
ratelimit-hour <ratelimithour> ratelimit-day <ratelimitday>
min-message-length <min-message-length value>
max-message-length <max-message-length value>
national-language-locking-shift <NationalLanguageIdentifier value>
national-language-single-shift <NationalLanguageIdentifier value>
```

#### DESCRIPTION

This command is used to modify the settings of an existing ESME configuration.

#### PARAMETERS

##### Standard Parameters

Name - The name of the ESME that is being modified.

##### Optional Parameters

Password - Specify the new password.  
It is used by the SMSC to authenticate the identity of

the binding ESME. The Service Provider may require ESMEs to provide a password when binding to the SMSC.

The new value takes effect when SMPP is restarted.

**networkId** - indicates virtual subnetwork that this ESME belongs. SMS flows within same networkId, unless changed using mproc (this is for multi-tenancy support). If this parameter is skipped - networkId will be set to "0" when ESME creation. If you do not use multi-tenancy support - set this value to 0 or skip.

**esme-ton** - Specify new ESME TON.  
If the SMPP Session Type is CLIENT, this TON will be used in the BIND request. If the SMPP Session Type is SERVER, the incoming BIND request should have the same TON as configured here. If the configured value is null (-1), SMSC will ignore it in both cases.

The new value takes effect when SMPP is restarted.

**esme-npi** - Specify new ESME NPI.  
If the SMPP Session Type is CLIENT, this NPI will be used in the BIND request. If the SMPP Session Type is SERVER, the incoming BIND request should have the same NPI as configured here. If the configured value is null (-1), SMSC will ignore it in both cases.

The new value takes effect when SMPP is restarted.

**esme-range** - Specify ESME Address Range.  
If the SMPP Session Type is CLIENT, this Address Range will be used in the BIND request. If the SMPP Session Type is SERVER, the incoming BIND request should have the same Address Range as configured here. If the configured value is null (-1), SMSC will ignore it in both cases.

The new value takes effect when SMPP is restarted.

**window-size** - Specify new window size.  
Default value is 1.  
The window size is the amount of unacknowledged requests that are permitted to be outstanding/unacknowledged at any given time. If more requests are added, the underlying stack will throw an exception.

This value is set only when ESME is defined as Client side. For Server side this value is taken from the 'SMPP Server Settings'.

The new value takes effect when SMPP is restarted.

`connect-timeout` - Default value is 10000 milli seconds.

This parameter is used to specify the time within which the connection to a remote SMSC server should be established.

This is useful only when ESME is defined as Client Side. For Server side this value is taken from the 'SMPP Server Settings'.

The new value takes effect when SMPP is restarted.

`request-expiry-timeout` - Default value is -1 (disabled).

This parameter is used to specify the time to wait in milli seconds for an endpoint to respond to before it expires.

This is useful only when ESME is defined as Client Side. For Server side this value is taken from the 'SMPP Server Settings'.

The new value takes effect when SMPP is restarted.

`window-monitor-interval` - Default value is -1 (disabled).

This parameter is used to specify the time between executions of monitoring the window for requests that expire. It is recommended that this value, generally, either matches or is half the value of 'request-expiry-timeout'. Therefore, in the worst case scenario, a request could take upto 1.5 times the 'requestExpiryTimeout' to clear out.

This is useful only when ESME is defined as Client Side. For Server side this value is taken from the 'SMPP Server Settings'.

The new value takes effect when SMPP is restarted.

`window-wait-timeout` - Default value is 60000 milli seconds.

This parameter is used to specify the time to wait until a slot opens up in the 'sendWindow'.

This is useful only when ESME is defined as Client Side. For Server side this value is taken from the 'SMPP Server Settings'.

The new value takes effect when SMPP is restarted.

`counters-enabled` - Default value is true.

When this is enabled, SMSC exposes the statistics for

SMPP connections.

This is useful only when ESME is defined as Client Side.  
For Server side this value is taken from the  
the 'SMPP Server Settings'.

The new value takes effect when SMPP is restarted.

**enquire-link-delay** - Default value is 30000 milli seconds.  
When SMSC connects to a remote server as CLIENT, it  
sends an 'ENQUIRE\_LINK' after every configured  
enquire-link-delay.

The new value takes effect immediately.

**charging-enabled** - Flag to enable or disable charging for every SMS  
arriving from SIP.

The new value takes effect immediately.

**source-ton** - Every SMS coming into the SMSC via this ESME should have  
the same 'source\_addr\_ton' as the value configured here.

If this configured value is null(-1) or not null and  
matches, the SMSC will compare the  
'source\_addr\_npi' and 'source\_addr\_range' as explained  
below.

If it doesn't match, the SMSC will reject this SMS with  
an error code '0x0000000A' indicating  
Invalid Source Address.

The new value takes effect immediately.

**source-npi** - Every SMS coming into the SMSC via this ESME should have  
the same 'source\_addr\_npi' as the value configured here.  
configured here.

If this configured value is null(-1)  
or not null and matches, the SMSC will compare the  
'source\_addr\_range' as below.

If it doesn't match, the SMSC will reject this  
SMS with an error code '0x0000000A' indicating  
Invalid Source Address.

The new value takes effect immediately.

**source-range** - Every SMS coming into the SMSC via this ESME should have  
the same 'source\_addr\_range' as the value configured  
here. This is a regular java expression and



default value is `^[0-9a-zA-Z]*`.

If it matches, the SMSC will accept the incoming SMS and process further.

If it doesn't match, the SMSC will reject this SMS with an error code `'0x0000000A'` indicating Invalid Source Address.

The new value takes effect immediately.

`routing-ton`      - The `DefaultSmsRoutingRule` will try to match the `'dest_addr_ton'` of outgoing SMS with the value configured here. If this configured value is `null(-1)` or not null and matches, the SMSC will compare the `'dest_addr_npi'` and `'destination_addr'` as explained below. If it doesn't match, the SMSC will select the next ESME in the list for matching routing rule.

`DefaultSmsRoutingRule` will consider ESME for routing only if

- 1) `SmppBindType` is `TRANSCEIVER`
- 2) `SmppBindType` is `RECEIVER` and `SmppSession.Type` is `SERVER`
- 3) `SmppBindType` is `TRANSMITTER` and `SmppSession.Type` is `CLIENT`

The new value takes effect immediately.

`routing-npi`      - The `DefaultSmsRoutingRule` will try to match the `'dest_addr_npi'` of outgoing SMS with the value configured here. If this configured value is `null(-1)` or not null and matches, the SMSC will compare the `'destination_addr'` as below. If it doesn't match, the SMSC will select the next ESME in the list for matching routing rule.

`DefaultSmsRoutingRule` will consider ESME for routing only if

- 1) `SmppBindType` is `TRANSCEIVER`
- 2) `SmppBindType` is `RECEIVER` and `SmppSession.Type` is `SERVER`
- 3) `SmppBindType` is `TRANSMITTER` and `SmppSession.Type` is `CLIENT`

The new value takes effect immediately.

`routing-range`      - The `DefaultSmsRoutingRule` will try to match the `'destination_addr'` of outgoing SMS with the value configured here. This is a regular java expression and default value is `^[0-9a-zA-Z]*`. If it matches, the SMSC

will send the SMS out over this SMPP connection. If it doesn't match, the SMSC will select the next ESME in the list for matching routing rule.

DefaultSmsRoutingRule will consider ESME for routing only if

- 1) SmppBindType is TRANSCEIVER
- 2) SmppBindType is RECEIVER and SmppSession.Type is SERVER
- 3) SmppBindType is TRANSMITTER and SmppSession.Type is CLIENT

The new value takes effect immediately.

**ratelimit\_second** - This parameter is used to specify a maximum limit of messages that the SMSC will accept from this ESME during any one second.

If the ESME sends more messages (per second) than the maximum limit specified by 'ratelimit\_second', these additional messages will be rejected by the SMSC GW along with an error code - "throttled".

The default value for this parameter is "0" and it implies "no restrictions". If this parameter is not specified it implies "no restrictions".

**ratelimit\_minute** - This parameter is used to specify a maximum limit of messages that the SMSC will accept from this ESME during any one minute.

If the ESME sends more messages (per minute) than the maximum limit specified by 'ratelimit\_minute', these additional messages will be rejected by the SMSC GW along with an error code - "throttled".

The default value for this parameter is "0" and it implies "no restrictions". If this parameter is not specified it implies "no restrictions".

**ratelimit\_hour** - This parameter is used to specify a maximum limit of messages that the SMSC will accept from this ESME during any one hour.

If the ESME sends more messages (per hour) than the maximum limit specified by 'ratelimit\_hour', these additional messages will be rejected by the SMSC GW along with an error code - "throttled".

The default value for this parameter is "0" and it implies "no restrictions". If this parameter is not

specified it implies "no restrictions".

`ratelimit_day` - This parameter is used to specify a maximum limit of messages that the SMSC will accept from this ESME during any one day.

If the ESME sends more messages (per day) than the maximum limit specified by 'ratelimit\_day', these additional messages will be rejected by the SMSC GW along with an error code - "throttled".

The default value for this parameter is "0" and it implies "no restrictions". If this parameter is not specified it implies "no restrictions".

`min-message-length` - This parameter is used to specify the minimum message length (in characters) acceptable to the SMSC GW, for messages coming from this ESME.

If an incoming message length is less than the min-message-length it will be rejected by SMSC GW.

The default value for this parameter is "-1" and it implies "no limitations". Any other negative value also implies "no limitations".

`max-message-length` - This parameter is used to specify the maximum message length (in characters) acceptable to the SMSC GW, for messages coming from this ESME.

If an incoming message length is more than the max-message-length it will be rejected by SMSC GW.

The default value for this parameter is "-1" and it implies "no limitations". Any other negative value also implies "no limitations".

#### SEE ALSO

`smc get scgt, smc set scgt, smc get scssn, smc set scssn, smc get hlrssn, smc set hlrssn, smc get mscssn, smc set mscssn, smc get maxmapv, smc set maxmapv, smpp esme create`

## Using GUI

### *Procedure: Modify an existing ESME using GUI*

1. In the GUI Management Console for SMSC Gateway, click on 'ESMEs' in the left panel.
2. The main panel will display the existing ESMEs (if any), one each in a row with corresponding actions (start, stop, delete, update) for each row.

3. You can update an existing by launching the 'ESME <name> properties' window by clicking on the blue coloured 'Update ESME' button. The 'ESME <name> properties' window will display all ESME parameters that can be updated by you. For more details of these parameters please refer to the descriptions of the CLI commands for the same in the preceding section.

ESME can be setup for SSL so every connection request should first do SSL hand-shake. Setting up SSL is only possible from GUI. After creating the ESME, users can edit property and enable SSL.



Only CLIENT ESME's (one that sends BIND request) can be enabled for SSL.

4. Update appropriate values for all the parameters and then click on the 'Close' button. This action will modify a new ESME with parameters as defined by you.
5. If there is an error in defining the ESME, then you will find the details of the error in the Management Console Log section below.

## View ESME Details

### Using CLI

You can view the details of all configured ESMEs by issuing the command `smpp esme show` as described below.

Name

`smpp esme show`

SYNOPSIS

`smpp esme show`

DESCRIPTION

This command is used to list all configured ESMEs.

### Using GUI

*Procedure: View ESME using the GUI*

1. In the GUI Management Console for SMSC Gateway, click on 'ESMEs' in the left panel.
2. The main panel will display the existing ESMEs (if any), one each in a row with corresponding actions (start, stop, delete) for each row.
3. You can view the details of an ESME by clicking on the row corresponding to the ESME. All relevant details of the ESME will be displayed in an expanded format.

# Delete an existing ESME

## Using CLI

You can delete any ESME by issuing the command `smpp esme delete` with appropriate parameters as described below.

### Name

`smpp esme delete`

### SYNOPSIS

`smpp esme delete <esmeName>`

### DESCRIPTION

This command is used to delete an existing ESME.

### PARAMETERS

`esmeName`            - Name of the ESME to be destroyed.

## Using GUI

*Procedure: Delete ESME using the GUI*

1. In the GUI Management Console for SMSC Gateway, click on 'ESMEs' in the left panel.
2. The main panel will display the existing ESMEs (if any), one each in a row with corresponding actions (start, stop, delete) for each row.
3. To delete an existing ESME click on the delete icon marked 'x' in red, for the row corresponding to the ESME. You can delete an ESME only if it is stopped.

# Start ESME

## Using CLI

You can start an ESME by issuing the command `smpp esme start` with appropriate parameters as described below.

#### Name

`smpp esme start`

#### SYNOPSIS

`smpp esme start <esmeName>`

#### DESCRIPTION

This command is used to start an existing ESME.

#### PARAMETERS

`esmeName`            - Name of the ESME to be started.

## Using GUI

*Procedure: Start ESME using the GUI*

1. In the GUI Management Console for SMSC Gateway, click on 'ESMEs' in the left panel.
2. The main panel will display the existing ESMEs (if any), one each in a row with corresponding actions (start, stop, delete) for each row.
3. To start an existing ESME click on the start icon lit in green, for the row corresponding to the ESME. You can start an ESME only if it is currently stopped.

## Stop ESME

## Using CLI

You can stop an ESME by issuing the command `smpp esme stop` with appropriate parameters as described below.

#### Name

`smpp esme stop`

#### SYNOPSIS

#### DESCRIPTION

This command is used to stop an already running ESME.

#### PARAMETERS

`esmeName`            - Name of the ESME to be stopped.

## Using GUI

*Procedure: Stop ESME using the GUI*

1. In the GUI Management Console for SMSC Gateway, click on 'ESMEs' in the left panel.

2. The main panel will display the existing ESMEs (if any), one each in a row with corresponding actions (start, stop, delete) for each row.
3. To stop an ESME click on the stop icon lit in red, for the row corresponding to the ESME. You can stop an ESME only if it is currently running.

## Other ESME Operations

### Using GUI

You can perform more operations in the GUI for any configured ESME. You can enable/disable Log Bytes and Log Pdu, dump window and reset counters.

*Procedure: Other ESME Operations using the GUI*

1. In the GUI Management Console for SMSC Gateway, click on 'ESMEs' in the left panel.
2. The main panel will display the existing ESMEs (if any), one each in a row with corresponding actions (start, stop, delete) for each row.
3. You can view the details of an ESME by clicking on the row corresponding to the ESME. All relevant details of the ESME will be displayed in an expanded format.
4. At the bottom of this expanded display you will find 6 buttons allowing you to perform the operations DisableLogBytes, DisableLogPdu, DumpWindow, EnableLogBytes, EnableLogPdu and ResetCounters.