Deployment Descriptor

ESME Hand Book

Copyright

Disclaimer

The information or statements given in this document concerning the suitability, capacity, or performance of the mentioned hardware or software products cannot be considered binding. Still, they shall be defined in the agreement made between 6d and the customer. However, 6d has made all reasonable efforts to ensure that the instructions contained in the document are adequate and free of material errors and omissions. 6d will, if necessary, explain issues, which may not be covered by the document.

Feedback

6d welcomes customer comments as a part of the process of continuous development and improvement of the documentation.

Trademarks and Registered Trademarks

Products and product names mentioned in this document may be trademarks or registered trademarks of their proprietors.

Revision History

Version	Issued Date	Serial Number	Description
1.0.	29-07-2020		Base Version
1.1	01-09-2020	6.6	Added one special case, ESME Other Language case
1.2	09-04-2021	6.3 , 6.4	Added two special cases, ELK Logging, ESME Receiving above 160 chars, in 6th section
		3.1	Added hyperlinks for connector and delivery connector sections in 3.1 section
		3.1	Added extra parameters in the 3.1 section starting from 13 - 21

		5	Added extra parameters in 5th section.
1.3	23-11-2021	6.1.1, 6.1.2	Added two new sub sections in 6.1 section
1.4	08-04-2022	6.6.1, 6.6.2	Added two new sub sections in 6.6 section.
1.5	20-04-2022	5	Updated 5th section with json and xml tags
		6 → 7 7 → 8	Renamed current 6th section as 7th and downwards
		6.1, 6.2	Introduced a new section – 6, for API Packets – both json & xml
		7.6	Added more details in 7.6
1.6	27-09-2022		Re-numberd few sections, Added missed details, Improvized submit & synch_submit packets

Table of Contents

1 Overview

The detailed description of the ESME (External Short Messaging Entity) module.

2 ESME Set Up

2.1 Wildfly setup basics

ESME should have a minimum wildfly 16 version along with JDK 1.8 for smooth running.

2.1.1Configuring jdk in wildfly

In the WildFly installation home directory (inside bin), we can find the *standalone.conf* (for UNIX based systems).

```
JAVA_HOME="/opt/java/jdk"
```

2.1.2Configuring Memory in Wildfly

In the WildFly installation home directory (inside bin), we can find the *standalone.conf* (for UNIX based systems).
We can simply open it and update the *-Xmx* option in the following line (change the number as per the requirement):

```
JAVA_OPTS="-Xms1024m-Xmx1024m"
```

This will set the heap memory while starting up the server.

2.1.3Configuring Standalone.xml

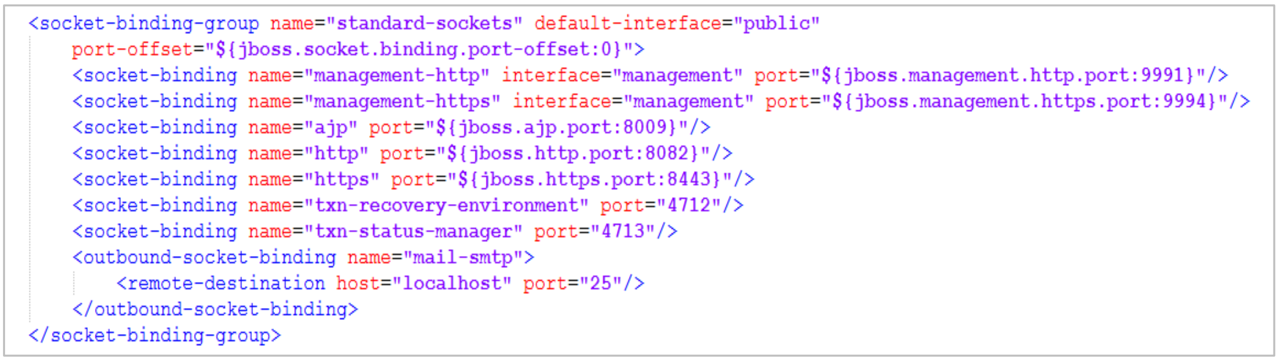
It can be seen in $WILDFLY_HOME/standalone/configuration/standalone.xml location.
Reach the <*socket-binding-group*> section to configure the following:

1. ajp port.
2. https port.
3. http port - default is 8080, we can change to any port available.

We can use netstat command to find port availability.

```
netstat -antp|grep 8080
```

4. management http port.
5. management https port.



```
<socket-binding-group name="standard-sockets" default-interface="public"
    port-offset="${jboss.socket.binding.port-offset:0}">
    <socket-binding name="management-http" interface="management" port="${jboss.management.http.port:9991}"/>
    <socket-binding name="management-https" interface="management" port="${jboss.management.https.port:9994}"/>
    <socket-binding name="ajp" port="${jboss.ajp.port:8009}"/>
    <socket-binding name="http" port="${jboss.http.port:8082}"/>
    <socket-binding name="https" port="${jboss.https.port:8443}"/>
    <socket-binding name="txn-recovery-environment" port="4712"/>
    <socket-binding name="txn-status-manager" port="4713"/>
    <outbound-socket-binding name="mail-smtp">
        <remote-destination host="localhost" port="25"/>
    </outbound-socket-binding>
</socket-binding-group>
```

Figure 1 Sample Socket-Binding-Group Configuration

2.1.4Deploying Esme.war

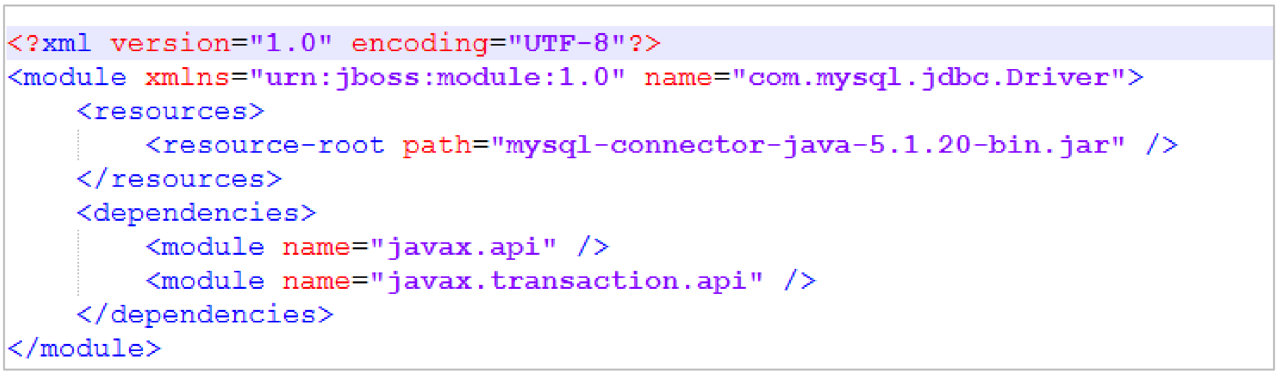
It consists of the following:

1. classes - consists of the configuration files such as log4j.xml and smpp.xml
2. lib - necessary library files
3. web.xml - definition and mapping of servlets

The two methods to deploy the Esme.war are: - via **admin panel** & via **dodeploy**. We have to keep a Esme.war.dodeploy marker file along with the Esme.war in $WILDFLY_HOME /standalone/deployments/ to trigger the deployment.

2.1.5Adding mysql/oracle jar to wildfly

Mysql/oracle jar must be added to the respective class path of the wildfly. In case of mysql, the path is like $WILDFLY_HOME/modules/system/layers/base/com/mysql/jdbc /Driver/main . We need to add the mysql jar and create the corresponding module.xml file. In module.xml, the module name should match the path name and the resource-root-path must be the name of the mysql jar.

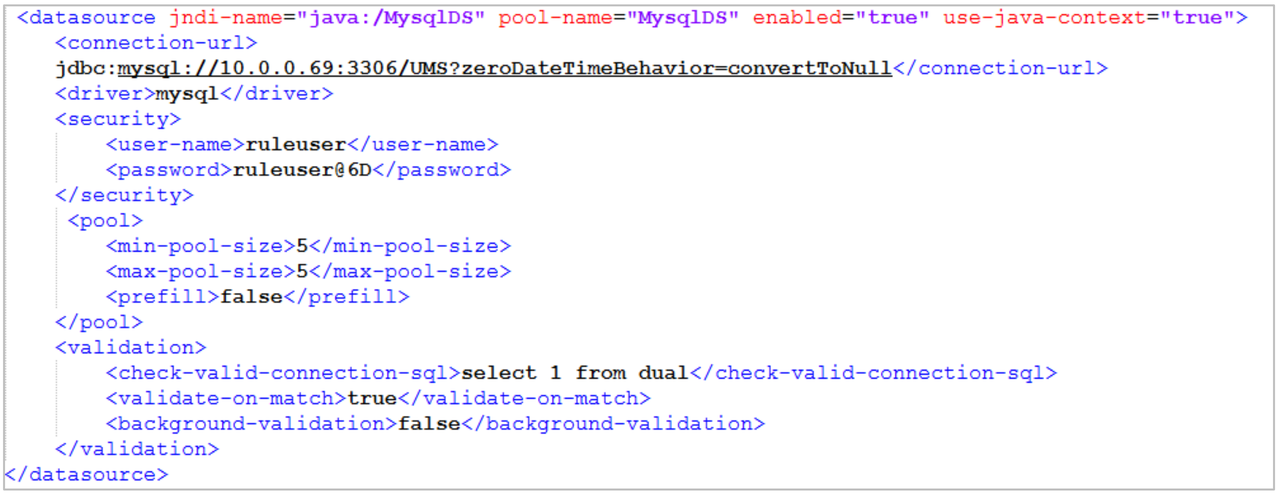


```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.0" name="com.mysql.jdbc.Driver">
    <resources>
        <resource-root path="mysql-connector-java-5.1.20-bin.jar" />
    </resources>
    <dependencies>
        <module name="javax.api" />
        <module name="javax.transaction.api" />
    </dependencies>
</module>
```

Figure 2 Module.xml

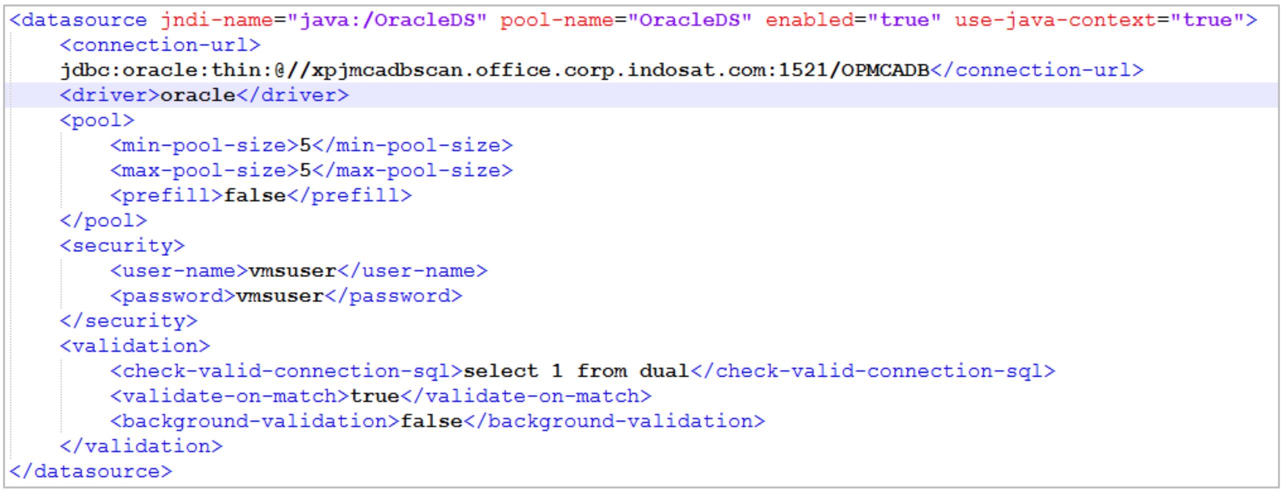
2.1.6Configuring jndi of mysql/oracle

We can configure the jndi of both mysql and oracle in the *datasources subsystem* in the standalone.xml file. This section contains the jndi name followed by connection -url, driver, security, pool and validation configurations.



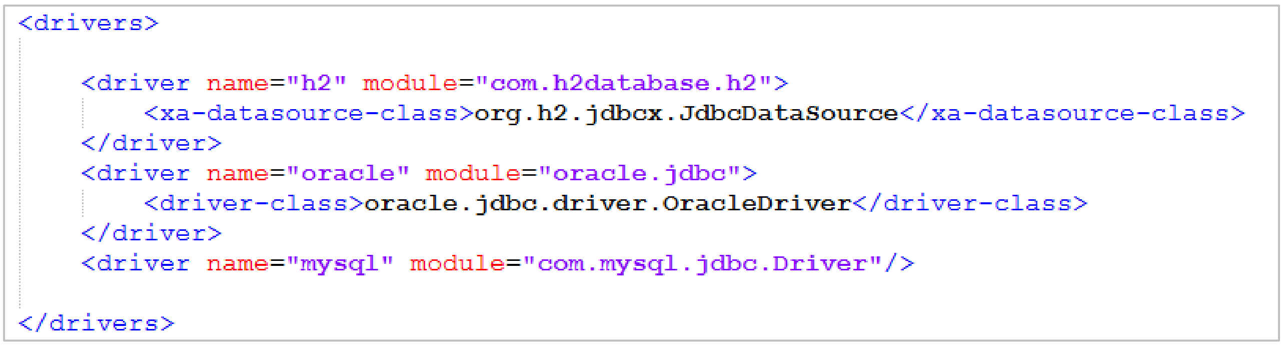
```
 <datasource jndi-name="java:/MysqlDS" pool-name="MysqlDS" enabled="true" use-java-context="true">
    <connection-url>
    jdbc:mysql://10.0.0.69:3306/UMS?zeroDateTimeBehavior=convertToNull</connection-url>
    <driver>mysql</driver>
    <security>
        <user-name>ruleuser</user-name>
        <password>ruleuser@6D</password>
    </security>
     <pool>
        <min-pool-size>5</min-pool-size>
        <max-pool-size>5</max-pool-size>
        <prefill>false</prefill>
    </pool>
    <validation>
        <check-valid-connection-sql>select 1 from dual</check-valid-connection-sql>
        <validate-on-match>true</validate-on-match>
        <background-validation>false</background-validation>
    </validation>
</datasource>
```

Figure 3 MySql Configuration



```
<datasource jndi-name="java:/OracleDS" pool-name="OracleDS" enabled="true" use-java-context="true">
    <connection-url>
    jdbc:oracle:thin:@//xpjmcadbscan.office.corp.indosat.com:1521/OPMCADB</connection-url>
    <driver>oracle</driver>
    <pool>
        <min-pool-size>5</min-pool-size>
        <max-pool-size>5</max-pool-size>
        <prefill>false</prefill>
    </pool>
    <security>
        <user-name>vmsuser</user-name>
        <password>vmsuser</password>
    </security>
    <validation>
        <check-valid-connection-sql>select 1 from dual</check-valid-connection-sql>
        <validate-on-match>true</validate-on-match>
        <background-validation>false</background-validation>
    </validation>
</datasource>
```

Figure 4 Oracle Configuration

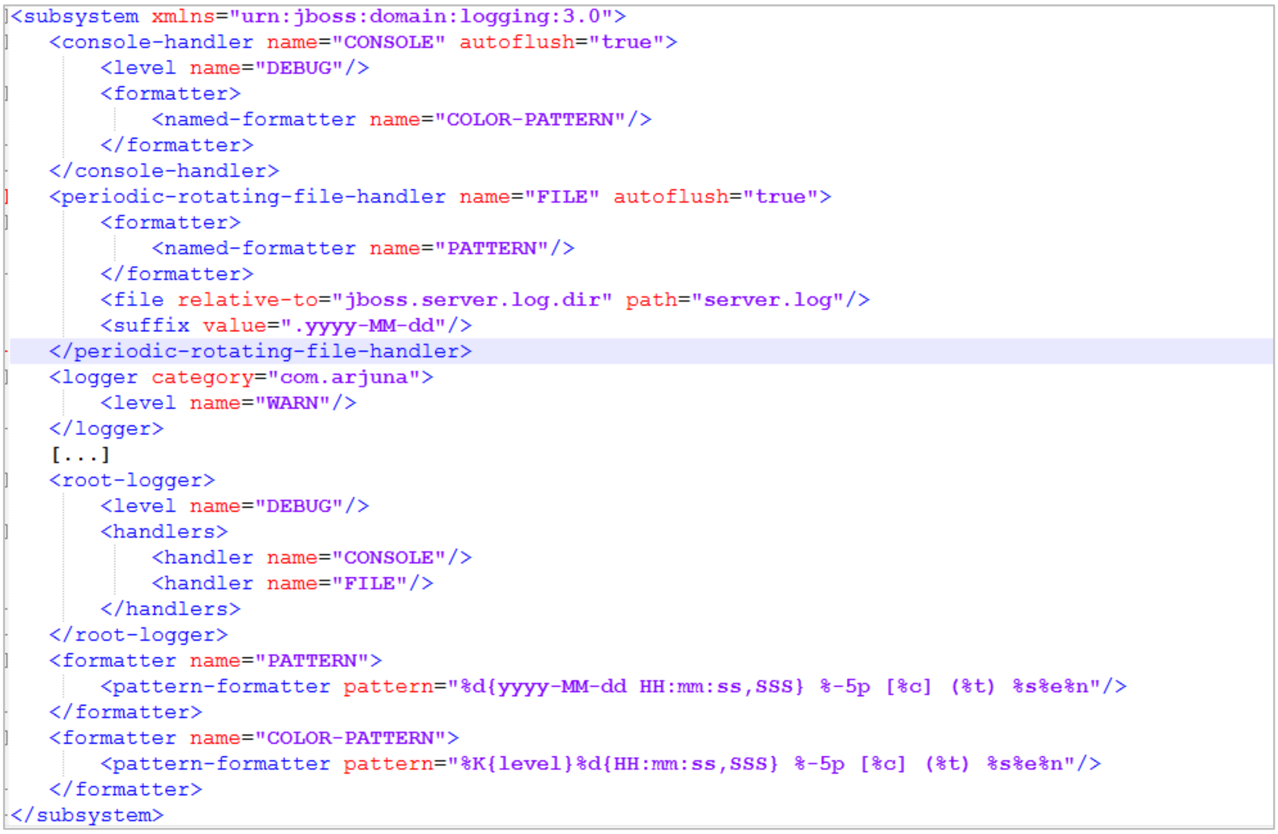


```
<drivers>

    <driver name="h2" module="com.h2database.h2">
        <xa-datasource-class>org.h2.jdbcx.JdbcDataSource</xa-datasource-class>
    </driver>
    <driver name="oracle" module="oracle.jdbc">
        <driver-class>oracle.jdbc.driver.OracleDriver</driver-class>
    </driver>
    <driver name="mysql" module="com.mysql.jdbc.Driver"/>

</drivers>
```

Figure 5 Driver Configurations

2.1.7Logging in wildfly

The overall server logging configuration is represented by the logging subsystem. It consists of four notable parts: handler configurations, logger, the root logger declarations and logging profiles.



```
<subsystem xmlns="urn:jboss:domain:logging:3.0">
    <console-handler name="CONSOLE" autoflush="true">
        <level name="DEBUG"/>
        <formatter>
            <named-formatter name="COLOR-PATTERN"/>
        </formatter>
    </console-handler>
    <periodic-rotating-file-handler name="FILE" autoflush="true">
        <formatter>
            <named-formatter name="PATTERN"/>
        </formatter>
        <file relative-to="jboss.server.log.dir" path="server.log"/>
        <suffix value=".yyyy-MM-dd"/>
    </periodic-rotating-file-handler>
    <logger category="com.arjuna">
        <level name="WARN"/>
    </logger>
    [...]
    <root-logger>
        <level name="DEBUG"/>
        <handlers>
            <handler name="CONSOLE"/>
            <handler name="FILE"/>
        </handlers>
    </root-logger>
    <formatter name="PATTERN">
        <pattern-formatter pattern="%d{yyyy-MM-dd HH:mm:ss,SSS} %-5p [%c] (%t) %s%e%n"/>
    </formatter>
    <formatter name="COLOR-PATTERN">
        <pattern-formatter pattern="%K{level}%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%e%n"/>
    </formatter>
</subsystem>
```

Figure 6 Logging Subsystem Configuration

Per-deployment logging allows you to add a logging configuration file to your deployment and have the logging for that deployment configured according to the configuration file. In a WAR deployment, the configuration file can be in WEB-INF/classes directories.
The following configuration files are allowed:

1. logging.properties.
2. jboss-logging.properties.
3. log4j.properties .
4. ***log4j.xml .***
5. jboss-log4j.xml.

3 ESME Application Configuration

3.1 Smpp.xml

The main task in manipulating our ESME is by ensuring that configurations in the smpp.xml are correct. It can be found at the $WILDFLY_HOME/standalone/ deployments/ESME.war/WEB-INF/classes/smpp.xml. A sample ***smpp.xml*** is attached here.
In order to create a successful binding with the SMSC (Short Message Service Centre), first make sure that following tags are configured correctly at both sides.

1. **smsc id** - unique smsc id
2. **IP address –** ip address of SMSC
3. **port** - smpp port
4. **System-id -** Identifies the ESME system requesting to bind as a transceiver with the SMSC
5. **Password** - As per the smpp standard, the password should be less than 9 characters.

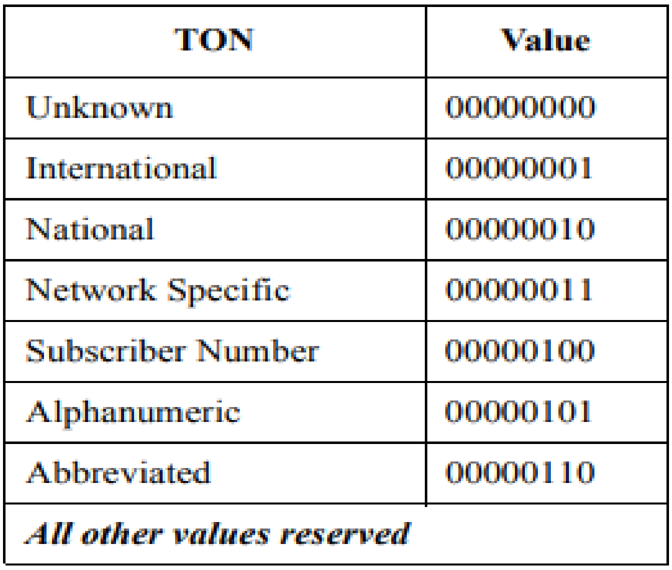
   We can get more ideas about this in the ***Smpp 3.4 Documentation*.** After making the initial configurations, we can proceed with the others.
6. **Bind mode** - The bind mode can be t, r, or tr for transmitter, receiver or transceiver respectively.
7. **Registereddelivery** - whether we need delivery receipt from smsc or not. The default setting of the registered_delivery parameter is 0x00.

```
<REGISTERED-DELIVERY>1</REGISTERED-DELIVERY>

1 → need delivery receipt          0 → no delivery receipt
```

8. **Interface-version** - should be equal to ascii value of the smpp version, current smpp version 3.4 ~ 4.
9. **Npi & Ton values** - the set of TON (Type of Number) & NPI (Numbering Plan Indicator) values.

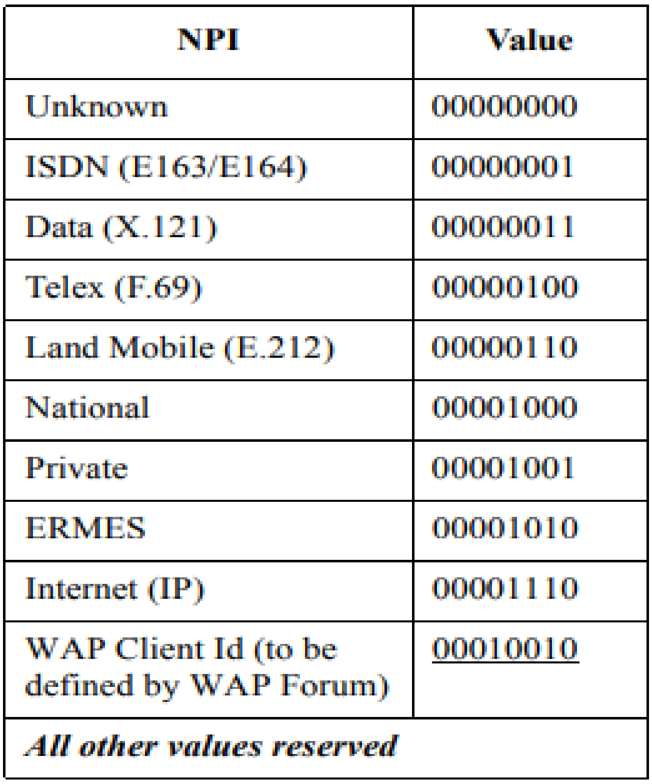
```
     <sourceAddrTon>0</sourceAddrTon>

     <sourceAddrNpi>0</sourceAddrNpi>

     <destAddrTon>0</destAddrTon>

     <destAddrNpi>0</destAddrNpi>
```

5 → Alphanumeric [address is alphanumeric, like a word with numbers]



TON	Value
Unknown	00000000
International	00000001
National	00000010
Network Specific	00000011
Subscriber Number	00000100
Alphanumeric	00000101
Abbreviated	00000110
All other values reserved	

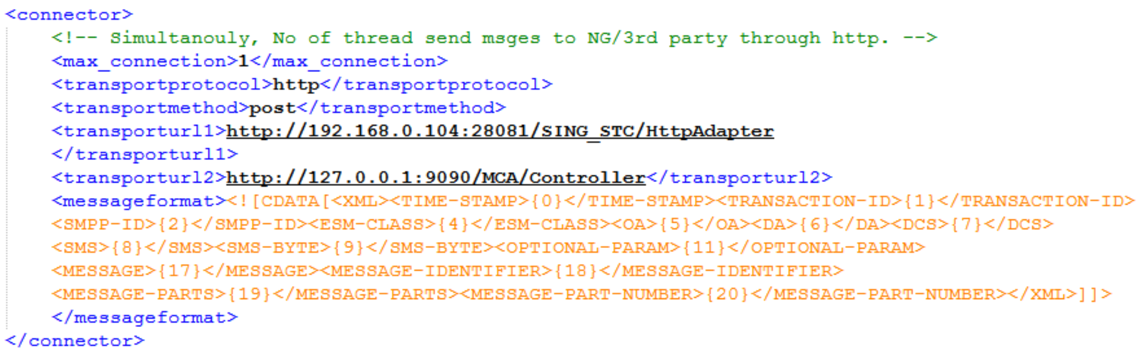
Table 5-3: TON values



NPI	Value
Unknown	00000000
ISDN (E163/E164)	00000001
Data (X.121)	00000011
Telex (F.69)	00000100
Land Mobile (E.212)	00000110
National	00001000
Private	00001001
ERMES	00001010
Internet (IP)	00001110
WAP Client Id (to be defined by WAP Forum)	00010010
All other values reserved	

Table 5-4: NPI values

10. **connector -** This section details the transport protocol, transport method, transport URLs and the message format we use to connect to the MO_Router or any other application in case of **any incoming feed from SMSC**. Refer in detail about the **connector section** of SMPP.xml here.



```
<connector>
    <!-- Simultanouly, No of thread send msges to NG/3rd party through http. -->
    <max_connection>1</max_connection>
    <transportprotocol>http</transportprotocol>
    <transportmethod>post</transportmethod>
    <transporturl1>http://192.168.0.104:28081/SING_STC/HttpAdapter
    </transporturl1>
    <transporturl2>http://127.0.0.1:9090/MCA/Controller</transporturl2>
    <messageformat><![CDATA[<XML><TIME-STAMP>{0}</TIME-STAMP><TRANSACTION-ID>{1}</TRANSACTION-ID>
    <SMPP-ID>{2}</SMPP-ID><ESM-CLASS>{4}</ESM-CLASS><OA>{5}</OA><DA>{6}</DA><DCS>{7}</DCS>
    <SMS>{8}</SMS><SMS-BYTE>{9}</SMS-BYTE><OPTIONAL-PARAM>{11}</OPTIONAL-PARAM>
    <MESSAGE>{17}</MESSAGE><MESSAGE-IDENTIFIER>{18}</MESSAGE-IDENTIFIER>
    <MESSAGE-PARTS>{19}</MESSAGE-PARTS><MESSAGE-PART-NUMBER>{20}</MESSAGE-PART-NUMBER></XML>]]>
    </messageformat>
</connector>
```

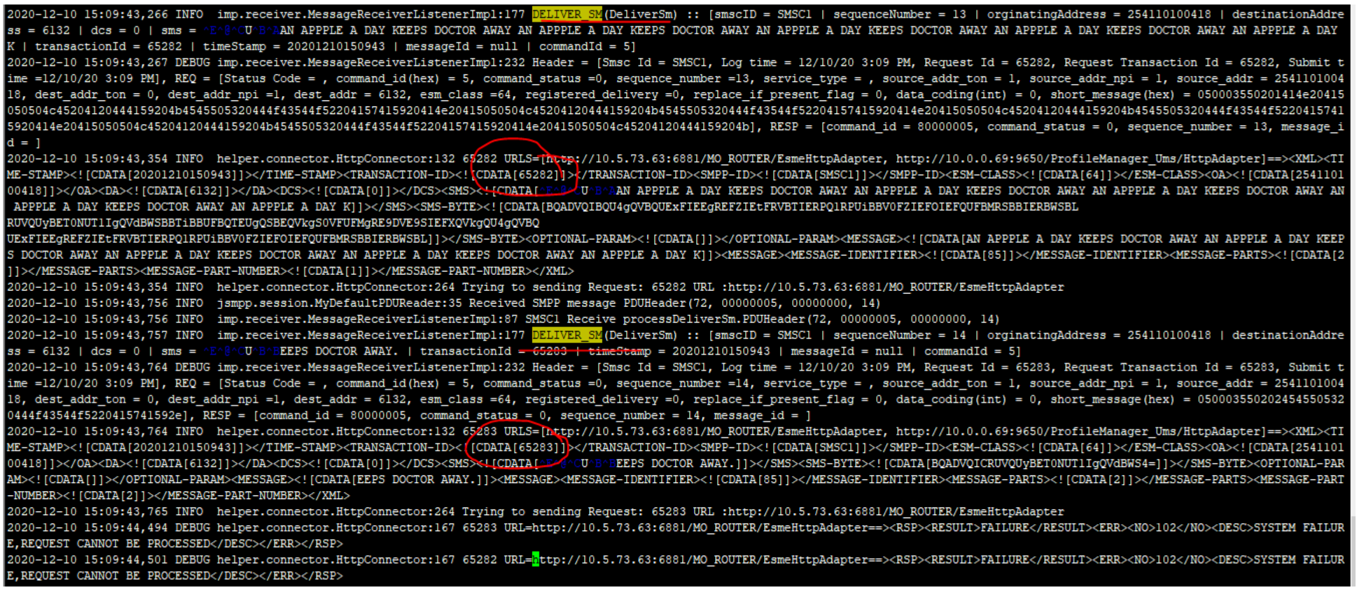
Figure 7 Connector XML Configuration

where

1. Max_connection - No. of thread that sends msgs to NG/3rd party through http. It can be of range 1-5 .
2. Transport Protocol - the required transport protocol like Http
3. Transport Method - the required transport method according to the corresponding transport protocol.
4. Transport URL - the URL to which the request xml should send.
5. Messageformat - the format in which request xml is sent from the ESME to the MOR. It behaves differently with the length of the message from the SMSC. [MOR]

Sample Message Format

1. When multi-part messages are coming from the SMSC, EsmClass 64



```
2020-12-10 15:09:43,266 INFO  imp.receiver.MessageReceiverListenerImpl:177 DELIVER_SM(DeliverSm) :: [smscID = SMSC1 | sequenceNumber = 13 | orginatingAddress = 254110100418 | destinationAddre
ss = 6132 | dcs = 0 | sms = ^E^@^CU^B^AAN APPPLE A DAY KEEPS DOCTOR AWAY AN APPPLE A DAY KEEPS DOCTOR AWAY AN APPPLE A DAY KEEPS DOCTOR AWAY AN APPPLE A DAY KEEPS DOCTOR AWAY AN APPPLE A DAY
K | transactionId = 65282 | timeStamp = 20201210150943 | messageId = null | commandId = 5]
2020-12-10 15:09:43,267 DEBUG imp.receiver.MessageReceiverListenerImpl:232 Header = [Smsc Id = SMSC1, Log time = 12/10/20 3:09 PM, Request Id = 65282, Request Transaction Id = 65282, Submit t
ime =12/10/20 3:09 PM], REQ = [Status Code = , command_id(hex) = 5, command_status =0, sequence_number =13, service_type = , source_addr_ton = 1, source_addr_npi = 1, source_addr = 2541101004
18, dest_addr_ton = 0, dest_addr_npi =1, dest_addr = 6132, esm_class =64, registered_delivery =0, replace_if_present_flag = 0, data_coding(int) = 0, short_message(hex) = 050003550201414e20415
050504c4520412044159204b4545505320444f43544f52204157415920414e20415050504c4520412044159204b4545505320444f43544f52204157415920414e20415050504c4520412044159204b4545505320444f43544f5220415741
5920414e20415050504c4520412044159204b4545505320444f43544f52204157415920414e20415050504c4520412044159204b], RESP = [command_id = 80000005, command_status = 0, sequence_number = 13, message_i
d = ]
2020-12-10 15:09:43,354 INFO  helper.connector.HttpConnector:132 65282 URLS=[http://10.5.73.63:6881/MO_ROUTER/EsmeHttpAdapter, http://10.0.0.69:9650/ProfileManager_Ums/HttpAdapter]==><XML><TI
ME-STAMP><![CDATA[20201210150943]]></TIME-STAMP><TRANSACTION-ID><![CDATA[65282]]></TRANSACTION-ID><SMPP-ID><![CDATA[SMSC1]]></SMPP-ID><ESM-CLASS><![CDATA[64]]></ESM-CLASS><OA><![CDATA[2541101
00418]]></OA><DA><![CDATA[6132]]></DA><DCS><![CDATA[0]]></DCS><SMS><![CDATA[^E^@^CU^B^AAN APPPLE A DAY KEEPS DOCTOR AWAY AN APPPLE A DAY KEEPS DOCTOR AWAY AN APPPLE A DAY KEEPS DOCTOR AWAY AN
 APPPLE A DAY KEEPS DOCTOR AWAY AN APPPLE A DAY K]]></SMS><SMS-BYTE><![CDATA[BQADVQIBQU4gQVBQUExFIEEgREFZIEtFRVBTIERPQ1RPUiBBV0FZIEFOIEFQUFBMRSBBIERBWSBL
RUVQUyBET0NUT1IgQVdBWSBBTiBBUFBQTEUgQSBEQVkgS0VFUFMgRE9DVE9SIEFXQVkgQU4gQVBQ
UExFIEEgREFZIEtFRVBTIERPQ1RPUiBBV0FZIEFOIEFQUFBMRSBBIERBWSBL]]></SMS-BYTE><OPTIONAL-PARAM><![CDATA[]]></OPTIONAL-PARAM><MESSAGE><![CDATA[AN APPPLE A DAY KEEPS DOCTOR AWAY AN APPPLE A DAY KEEP
S DOCTOR AWAY AN APPPLE A DAY KEEPS DOCTOR AWAY AN APPPLE A DAY KEEPS DOCTOR AWAY AN APPPLE A DAY K]]><MESSAGE><MESSAGE-IDENTIFIER><![CDATA[85]]></MESSAGE-IDENTIFIER><MESSAGE-PARTS><![CDATA[2
]]></MESSAGE-PARTS><MESSAGE-PART-NUMBER><![CDATA[1]]></MESSAGE-PART-NUMBER></XML>
2020-12-10 15:09:43,354 INFO  helper.connector.HttpConnector:264 Trying to sending Request: 65282 URL :http://10.5.73.63:6881/MO_ROUTER/EsmeHttpAdapter
2020-12-10 15:09:43,756 INFO  jsmpp.session.MyDefaultPDUReader:35 Received SMPP message PDUHeader(72, 00000005, 00000000, 14)
2020-12-10 15:09:43,756 INFO  imp.receiver.MessageReceiverListenerImpl:87 SMSC1 Receive processDeliverSm.PDUHeader(72, 00000005, 00000000, 14)
2020-12-10 15:09:43,757 INFO  imp.receiver.MessageReceiverListenerImpl:177 DELIVER_SM(DeliverSm) :: [smscID = SMSC1 | sequenceNumber = 14 | orginatingAddress = 254110100418 | destinationAddre
ss = 6132 | dcs = 0 | sms = ^E^@^CU^B^BEEPS DOCTOR AWAY. | transactionId = 65283 | timeStamp = 20201210150943 | messageId = null | commandId = 5]
2020-12-10 15:09:43,764 DEBUG imp.receiver.MessageReceiverListenerImpl:232 Header = [Smsc Id = SMSC1, Log time = 12/10/20 3:09 PM, Request Id = 65283, Request Transaction Id = 65283, Submit t
ime =12/10/20 3:09 PM], REQ = [Status Code = , command_id(hex) = 5, command_status =0, sequence_number =14, service_type = , source_addr_ton = 1, source_addr_npi = 1, source_addr = 2541101004
18, dest_addr_ton = 0, dest_addr_npi =1, dest_addr = 6132, esm_class =64, registered_delivery =0, replace_if_present_flag = 0, data_coding(int) = 0, short_message(hex) = 050003550202454550532
0444f43544f5220415741592e], RESP = [command_id = 80000005, command_status = 0, sequence_number = 14, message_id = ]
2020-12-10 15:09:43,764 INFO  helper.connector.HttpConnector:132 65283 URLS=[http://10.5.73.63:6881/MO_ROUTER/EsmeHttpAdapter, http://10.0.0.69:9650/ProfileManager_Ums/HttpAdapter]==><XML><TI
ME-STAMP><![CDATA[20201210150943]]></TIME-STAMP><TRANSACTION-ID><![CDATA[65283]]></TRANSACTION-ID><SMPP-ID><![CDATA[SMSC1]]></SMPP-ID><ESM-CLASS><![CDATA[64]]></ESM-CLASS><OA><![CDATA[2541101
00418]]></OA><DA><![CDATA[6132]]></DA><DCS><![CDATA[0]]></DCS><SMS><![CDATA[^E^@^CU^B^BEEPS DOCTOR AWAY.]]></SMS><SMS-BYTE><![CDATA[BQADVQICRUVQUyBET0NUT1IgQVdBWS4=]]></SMS-BYTE><OPTIONAL-PAR
AM><![CDATA[]]></OPTIONAL-PARAM><MESSAGE><![CDATA[EEPS DOCTOR AWAY.]]><MESSAGE><MESSAGE-IDENTIFIER><![CDATA[85]]></MESSAGE-IDENTIFIER><MESSAGE-PARTS><![CDATA[2]]></MESSAGE-PARTS><MESSAGE-PART
-NUMBER><![CDATA[2]]></MESSAGE-PART-NUMBER></XML>
2020-12-10 15:09:43,765 INFO  helper.connector.HttpConnector:264 Trying to sending Request: 65283 URL :http://10.5.73.63:6881/MO_ROUTER/EsmeHttpAdapter
2020-12-10 15:09:44,494 DEBUG helper.connector.HttpConnector:167 65283 URL=http://10.5.73.63:6881/MO_ROUTER/EsmeHttpAdapter==><RSP><RESULT>FAILURE</RESULT><ERR><NO>102</NO><DESC>SYSTEM FAILUR
E,REQUEST CANNOT BE PROCESSED</DESC></ERR></RSP>
2020-12-10 15:09:44,501 DEBUG helper.connector.HttpConnector:167 65282 URL=http://10.5.73.63:6881/MO_ROUTER/EsmeHttpAdapter==><RSP><RESULT>FAILURE</RESULT><ERR><NO>102</NO><DESC>SYSTEM FAILUR
E,REQUEST CANNOT BE PROCESSED</DESC></ERR></RSP>
```

Connector xml when multi-part messages come from SMSC

We could see the long message from SMSC is getting splitted into 2 multi-part messages .

<MESSAGE> have the decoded message string from **<SMS-BYTE>** tag, after removing UDH headers. If you compare the **<SMS>** with **<MESSAGE>**, we can see that there are few unwanted characters coming in front of the message in **<SMS>**. Those are UDH headers in unreadable format.

<MESSAGE-IDENTIFIER> gives the message identifier, here 85 that is coming in both packets of multi-part messages. This identifier is used to concat these splitted messages at the MOR side.

<MESSAGE-PARTS> gives the total number of Message Parts. Here, it is two, which means two part-messages will be there.

<MESSAGE-PART-NUMBER> gives the sequence number of Message Part. Here, we could see in the first packet, it is given as one, and the second packet, it is given as two. This helps to find the order in which we need to concat the part-messages.

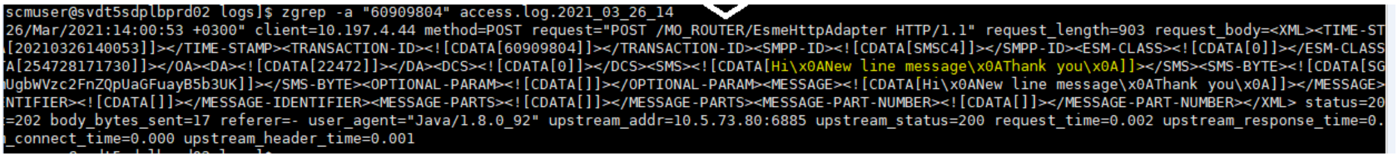
2. When single message [less than 160chars] are coming, EsmClass 0

<XML>
<TIME-STAMP><![CDATA[20210409152040]]></TIME-STAMP>
<TRANSACTION-ID><![CDATA[49534]]></TRANSACTION-ID>
<SMPP-ID><![CDATA[airtel]]></SMPP-ID>
<ESM-CLASS><![CDATA[0]]></ESM-CLASS>
<OA><![CDATA[4477665544]]></OA>
<DA><![CDATA[337788665522]]></DA>
<DCS><![CDATA[0]]></DCS>
<SMS><![CDATA[Hello from SMPPSim]]></SMS>
<SMS-BYTE><![CDATA[SGVsbG8gZnJvbSBTTVBQU2lt]]></SMS-BYTE>
<OPTIONAL-PARAM><![CDATA[]]></OPTIONAL-PARAM>
<MESSAGE><![CDATA[Hello from SMPPSim]]></MESSAGE>
<MESSAGE-IDENTIFIER><![CDATA[]]></MESSAGE-IDENTIFIER>
<MESSAGE-PARTS><![CDATA[]]></MESSAGE-PARTS>
<MESSAGE-PART-NUMBER><![CDATA[]]></MESSAGE-PART-NUMBER>
</XML>

MESSAGE → it has the same message from the <SMS>, since esm class value is 0 and no multi-part messages are coming. No UDH headers are coming from SMSC .

MESSAGE-IDENTIFIER, MESSAGE-PARTS, MESSAGE-PART-NUMBER → Empty since No UDH headers are coming from SMSC.

3. When new line character is coming from SMSC



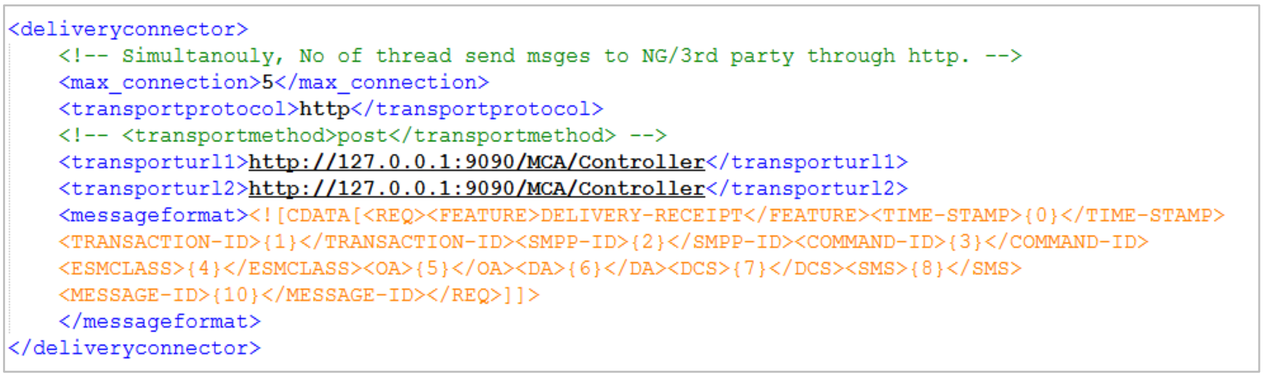
```
scmuser@svdt5sdplbprd02 logs]$ zgrep -a "60909804" access.log.2021_03_26_14
26/Mar/2021:14:00:53 +0300" client=10.197.4.44 method=POST request="POST /MO_ROUTER/EsmeHttpAdapter HTTP/1.1" request_length=903 request_body=<XML><TIME-ST
[20210326140053]]></TIME-STAMP><TRANSACTION-ID><![CDATA[60909804]]></TRANSACTION-ID><SMPP-ID><![CDATA[SMSC4]]></SMPP-ID><ESM-CLASS><![CDATA[0]]></ESM-CLASS
A[254728171730]]></OA><DA><![CDATA[22472]]></DA><DCS><![CDATA[0]]></DCS><SMS><![CDATA[Hi\x0ANew line message\x0AThank you\x0A]]></SMS><SMS-BYTE><![CDATA[SG
UgbWVzc2FnZQpUaGFuayB5b3UK]]></SMS-BYTE><OPTIONAL-PARAM><![CDATA[]]></OPTIONAL-PARAM><MESSAGE><![CDATA[Hi\x0ANew line message\x0AThank you\x0A]]></MESSAGE>
NTIFIER><![CDATA[]]></MESSAGE-IDENTIFIER><MESSAGE-PARTS><![CDATA[]]></MESSAGE-PARTS><MESSAGE-PART-NUMBER><![CDATA[]]></MESSAGE-PART-NUMBER></XML> status=20
=202 body_bytes_sent=17 referer=- user_agent="Java/1.8.0_92" upstream_addr=10.5.73.80:6885 upstream_status=200 request_time=0.002 upstream_response_time=0.
_connect_time=0.000 upstream_header_time=0.001
```

Sample packet with new line character

Placeholder	Message Format & Description
0	TimeStamp
1	TransactionId
2	SmscID
3	CommandId
4	EsmClass
5	Originating Address
6	Destination Address
7	Dcs
8	Sms
9	Sms Byte
10	Message Id
11	Optional Param Text
12	Command Status
13	Sequence Number
14	Hexadecimal sms Byte
15	Registered Delivery
16	Service Type
17	If EsmClass is 64, Sms data without UDH headers or else just Sms Message
18	If EsmClass is 64, MESSAGE-IDENTIFIER value from UDH or else empty
19	If EsmClass is 64, total number of MESSAGE-PARTS in UDH or else empty
20	If EsmClass is 64, MESSAGE-PART-NUMBER from UDH or else empty

Figure 8 Messageformat tag names and Descriptions

11. **Delivery connector -** This section details the transport protocol, transport method, transport URLs and the message format we use to **send the delivery receipt** , when **RegisteredDelivery is 1**, to the required 3rd party module like UMS/NG. Refer in detail about **delivery connector** of SMPP.xml here.



```
<deliveryconnector>
    <!-- Simultanouly, No of thread send msges to NG/3rd party through http. -->
    <max_connection>5</max_connection>
    <transportprotocol>http</transportprotocol>
    <!-- <transportmethod>post</transportmethod> -->
    <transporturl1>http://127.0.0.1:9090/MCA/Controller</transporturl1>
    <transporturl2>http://127.0.0.1:9090/MCA/Controller</transporturl2>
    <messageformat><![CDATA[<REQ><FEATURE>DELIVERY-RECEIPT</FEATURE><TIME-STAMP>{0}</TIME-STAMP>
    <TRANSACTION-ID>{1}</TRANSACTION-ID><SMPP-ID>{2}</SMPP-ID><COMMAND-ID>{3}</COMMAND-ID>
    <ESMCLASS>{4}</ESMCLASS><OA>{5}</OA><DA>{6}</DA><DCS>{7}</DCS><SMS>{8}</SMS>
    <MESSAGE-ID>{10}</MESSAGE-ID></REQ>]]>
    </messageformat>
</deliveryconnector>
```

Figure 9 Delivery Connector XML Configuration

Placeholder	Tagname & Description
0	TimeStamp
1	TransactionId
2	SmscID
3	CommandId
4	Esm Class
5	Originating Address
6	Destination Address
7	Dcs
8	Sms
9	SmsByte
10	MessageId
11	OptionalParamText
12	CommandStatus
13	SequenceNumber
14	Hexadecimal smsByte
15	RegisteredDelivery
16	ServiceType

Figure 10 Tagname and Descriptions

Sample Message Format

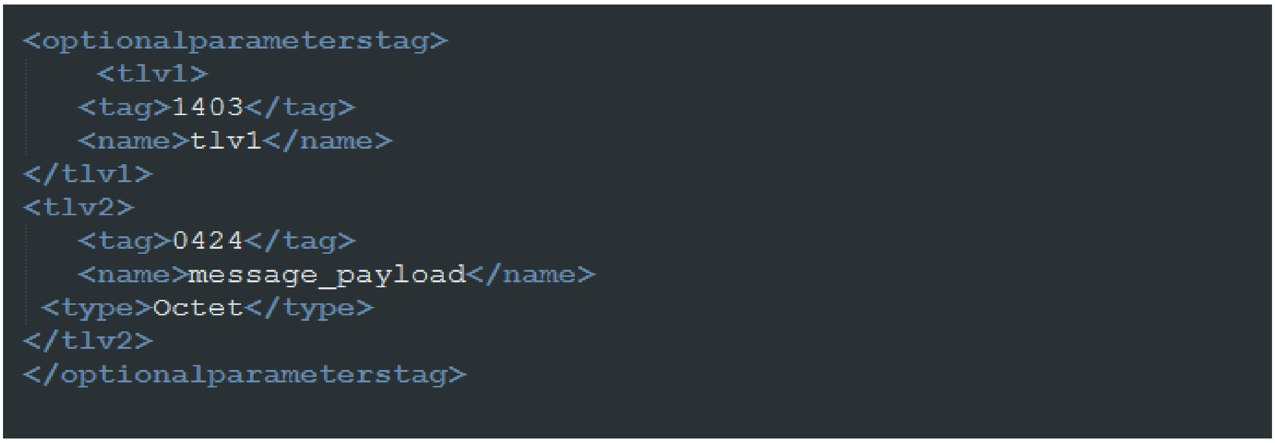
```
<REQ>
<FEATURE>DELIVERY-RECEIPT</FEATURE>
<TIME-STAMP><![CDATA[20210409160127]]></TIME-STAMP>
<TRANSACTION-ID><![CDATA[13600]]></TRANSACTION-ID>
<SMPP-ID><![CDATA[airtel]]></SMPP-ID>
<COMMAND-ID><![CDATA[5]]></COMMAND-ID>
<ESM-CLASS><![CDATA[4]]></ESM-CLASS>
```

```
<OA><![CDATA[9611726551]]></OA>
<DA><![CDATA[8688492301]]></DA>
<DCS><![CDATA[2]]></DCS>
<SMS><![CDATA[id:0 sub:001 dlvrd:001 submit date:2104091601 done date:2104091601
stat:DELIVRD err:000 Text:I called You at Wed,]]></SMS>
<MESSAGE-ID><![CDATA[0]]></MESSAGE-ID>
</REQ>
```

Where

FEATURE → here feature will be DELIVERY-RECEIPT
ESM-CLASS → the value of **EsmClass will be 4, for delivery-receipt**

12. **Optional parameters -** these are the optional tags that can be used in special cases like message payload. (Refer to the section about ***special cases***).



```
<optionalparameterstag>
    <tlv1>
   <tag>1403</tag>
   <name>tlv1</name>
</tlv1>
<tlv2>
   <tag>0424</tag>
   <name>message_payload</name>
 <type>Octet</type>
</tlv2>
</optionalparameterstag>
```

Figure 11 Optional Parameters Configuration

13. **Service-type** - The service_type parameter can be used to indicate the SMS Application service associated with the message.

Specifying the service_type allows the ESME to

- avail of enhanced messaging services such as replace by service type
- to control the tele-service used on the air interface.

Set to NULL for default SMSC settings.

14. **Dcs** - It is known as Data coding scheme. Defines the encoding scheme of the short message user data. Zero is the default, for English. For other languages, dcs value used is 8. For Flash Message, dcs value can be 16.

15. **messageReceiverListenerImpl** - This tag gives the full class path to the class messageReceiverListenerImpl. Current value is, *com.sixdee.imp.receiver.MessageReceiverListenerImpl* . Don't change this value unless Development Team suggests so.

16. **TPS** - Transaction per second (TPS) is the number of transactions executed per second. In other words, it can be calculated based on how many transactions are executed over a certain duration of the test and then calculate it for a second.For

example, if a user executes 6 transactions every minute, then the TPS would be 6 transactions/60 sec = 0.10 TPS.

The default value will be -1. Change to 1 when ESME is going to be implemented with retry.

17. **enquireLinkTimer** - The enquire link timer is an SMPP session timer to enable the ESME request the SMPP session status of the other communicating SMPP entity via the *enquire_link* command. The value ranges from 5 - 10, in seconds.

18. **receive-timeout** - The receive timeout is a timeout for trying to receive a message from the SMSC in synchronize way. Set the receive-timeout to number of seconds. The value ranges from 5 - 10, in seconds.

19. **pdu-processors** - No of thread process simultaneously for SMSC request. Max value can be 5.

20. **pdu-maxQueueSize** - set the pdu-processor waiting queue size. If you want go to infinite, set the pdu-maxQueueSize to -1.Otherwise set the pdu-maxQueueSize to number .

21. **allowedNumSeries** - This tag lists the allowed number series. Default value is “*”, denoting “all” .

4 API Packets

4.1 ESME Xml Request Packet

Adapters: http://localhost:8081/ESME/HttpAdapter
http://localhost:8081/ESME/HttpNoRespAdapter

4.1.1 Submit_sm

```
<REQ>
  <FEATURE>submit_sm</FEATURE>
  <TIME-STAMP>25062009103510</TIME-STAMP>
  <RESPONSE-URL>http://localhost:8080/ESME/TestServlet</RESPONSE-URL>
  <PARAMETERS>
  <REQ-TRANSACTION-ID>0</REQ-TRANSACTION-ID>
  <SMSC-ID>airtel</SMSC-ID>
  <SUBMIT-SM>
  <SOURCE-ADDR-TON>0</SOURCE-ADDR-TON>
  <SOURCE-ADDR-NPI>0</SOURCE-ADDR-NPI>
  <SOURCE-ADDR>1234</SOURCE-ADDR>
  <DEST-ADDR-TON>0</DEST-ADDR-TON>
  <DEST-ADDR-NPI>0</DEST-ADDR-NPI>
  <DESTINATION-ADDR>9945626828</DESTINATION-ADDR>
<ESM-CLASS>0</ESM-CLASS>
 <concatenatedSMesmClass>0</concatenatedSMesmClass>
  <SHORT-MESSAGE>UkNIUiAyMjIy</SHORT-MESSAGE>
  <DATA-CODING>0</DATA-CODING>
  </SUBMIT-SM>
  </PARAMETERS>
 </REQ>
```

Instant response

```
<RESP>
    <TIME-STAMP>23072021202222</TIME-STAMP>
    <STATUS>0</STATUS>
</RESP>
```

Final response

```
<RESP>
   <STATUS>0</STATUS>
   <STATUS-CODE>0</STATUS-CODE>
   <TIME-STAMP>30072021192109</TIME-STAMP>
   <PARAMETERS>
      <REQ-TRANSACTION-ID>0</REQ-TRANSACTION-ID>
      <RESP-TRANSACTION-ID>0</RESP-TRANSACTION-ID>
      <SMSC-ID>airtel</SMSC-ID>
   </PARAMETERS>
</RESP>
```

4.1.2 Synch_Submit_sm

```
<REQ>
 <FEATURE>synch_submit_sm</FEATURE>
 <TIME-STAMP>09092015184542</TIME-STAMP>
 <PARAMETERS>
 <REQ-TRANSACTION-ID>2</REQ-TRANSACTION-ID>
 <SMSC-ID>airtel</SMSC-ID>
 <SUBMIT-SM>
 <SOURCE-ADDR-TON>0</SOURCE-ADDR-TON>
 <SOURCE-ADDR-NPI>0</SOURCE-ADDR-NPI>
 <SOURCE-ADDR>8688492301</SOURCE-ADDR>
 <DEST-ADDR-TON>0</DEST-ADDR-TON>
 <DEST-ADDR-NPI>0</DEST-ADDR-NPI>
 <DESTINATION-ADDR>222222222</DESTINATION-ADDR>
 <ESM-CLASS>0</ESM-CLASS>
 <concatenatedSMesmClass>0</concatenatedSMesmClass>
 <SHORT-MESSAGE>SSBjYWxsZWQgWW91IGF0IFdlZCwgOSBTZXAgMjAxNSAxODoxMDoyNQ==</SHORT-MESSAGE>
  <REGISTERED-DELIVERY>1</REGISTERED-DELIVERY>
<DATA-CODING>0</DATA-CODING>
 </SUBMIT-SM>
 </PARAMETERS>
 </REQ>
```

Synch_submit_response

```
<RESP>
    <STATUS>0</STATUS>
    <STATUS-CODE>0</STATUS-CODE>
    <TIME-STAMP>23072021202301</TIME-STAMP>
    <PARAMETERS>
        <REQ-TRANSACTION-ID>2</REQ-TRANSACTION-ID>
        <RESP-TRANSACTION-ID>7</RESP-TRANSACTION-ID>
        <SMSC-ID>airtel</SMSC-ID>
        <MESSAGE-ID>5</MESSAGE-ID>
    </PARAMETERS>
</RESP>
```

4.1.3 MESSAGE PAYLOAD & UDH

4.1.3.1 UDH Support [Primary Method]

```
<REQ>
 <FEATURE>submit_sm</FEATURE>
 <TIME-STAMP>09092015184542</TIME-STAMP>
 <PARAMETERS>
 <REQ-TRANSACTION-ID>2</REQ-TRANSACTION-ID>
 <SMSC-ID>airtel</SMSC-ID>
 <SUBMIT-SM>
 <SOURCE-ADDR-TON>0</SOURCE-ADDR-TON>
 <SOURCE-ADDR-NPI>0</SOURCE-ADDR-NPI>
 <SOURCE-ADDR>8688492301</SOURCE-ADDR>
 <DEST-ADDR-TON>0</DEST-ADDR-TON>
 <DEST-ADDR-NPI>0</DEST-ADDR-NPI>
 <DESTINATION-ADDR>222222222</DESTINATION-ADDR>
 <ESM-CLASS>64</ESM-CLASS>
 <concatenatedSMesmClass>64</concatenatedSMesmClass>
 <SHORT-
MESSAGE>SSBjYWxsZWQgWW91IGF0IFdlZCwgOSBTZXAgMjAxNSAxODoxMDo
yNQ==</SHORT-MESSAGE>
<DATA-CODING>0</DATA-CODING>
 </SUBMIT-SM>
 </PARAMETERS>
 </REQ>
```

4.1.3.2 Message Payload [SMSC Support Required]

```
<REQ>
 <FEATURE>submit_sm</FEATURE>
 <TIME-STAMP>09092015184542</TIME-STAMP>
 <PARAMETERS>
 <REQ-TRANSACTION-ID>2</REQ-TRANSACTION-ID>
 <SMSC-ID>airtel</SMSC-ID>
 <SUBMIT-SM>
 <SOURCE-ADDR-TON>0</SOURCE-ADDR-TON>
 <SOURCE-ADDR-NPI>0</SOURCE-ADDR-NPI>
 <SOURCE-ADDR>8688492301</SOURCE-ADDR>
 <DEST-ADDR-TON>0</DEST-ADDR-TON>
 <DEST-ADDR-NPI>0</DEST-ADDR-NPI>
 <DESTINATION-ADDR>222222222</DESTINATION-ADDR>
 <ESM-CLASS>0</ESM-CLASS>
 <concatenatedSMesmClass>0</concatenatedSMesmClass>
 <SHORT-
MESSAGE>SSBjYWxsZWQgWW91IGF0IFdlZCwgOSBTZXAgMjAxNSAxODoxMDo
yNQ==</SHORT-MESSAGE>
<DATA-CODING>0</DATA-CODING>
<MESSAGE-PAYLOAD>true</MESSAGE-PAYLOAD>
 </SUBMIT-SM>
 </PARAMETERS>
 </REQ>
```

4.1.4ESME MO & Delivery Packets [in smpp.xml]

4.1.4.1 MO Flow [Connector url]

```
<XML>
<TIME-STAMP>{0}</TIME-STAMP>
<TRANSACTION-ID>{1}</TRANSACTION-ID>
<SMPP-ID>{2}</SMPP-ID>
```

```
<ESM-CLASS>{4}</ESM-CLASS>
<OA>{5}</OA>
<DA>{6}</DA>
<DCS>{7}</DCS>
<SMS>{8}</SMS>
<SMS-BYTE>{9}</SMS-BYTE>
<OPTIONAL-PARAM>{11}</OPTIONAL-PARAM>
<MESSAGE>{17}</MESSAGE>
<MESSAGE-IDENTIFIER>{18}</MESSAGE-IDENTIFIER>
<MESSAGE-PARTS>{19}</MESSAGE-PARTS>
<MESSAGE-PART-NUMBER>{20}</MESSAGE-PART-NUMBER>
</XML>
```

Sample

```
<XML>
<TIME-STAMP><![CDATA[20210730194044]]></TIME-STAMP>
<TRANSACTION-ID><![CDATA[74376]]></TRANSACTION-ID>
<SMPP-ID><![CDATA[airtel]]></SMPP-ID>
<OA><![CDATA[4477665544]]></OA>
<DA><![CDATA[337788665522]]></DA>
<DCS><![CDATA[0]]></DCS>
<SMS><![CDATA[Hello from SMPPSim Hello from SMPPSim Hello from SMPPSim
Hello from SMPPSimHello from SMPPSimHello from SMPPSimHello from
SMPPSim]]></SMS>
<SMS-
BYTE><![CDATA[IEhlbGxvIGZyb20gU01QUFNpbSBIZWxsbyBmcm9tIFNNUFBTaW
0gSGVsbG8gZnJvbSBTTVBQU2ltIEhlbGxvIGZyb20gU01QUFNpbUhlbGxvIGZyb20
gU01QUFNpbUhlbGxvIGZyb20gU01QUFNpbUhlbGxvIGZyb20gU01QUFNpbQ==]]>
</SMS-BYTE>
<OPTIONAL-PARAM><![CDATA[]]></OPTIONAL-PARAM>
<MESSAGE><![CDATA[from SMPPSim Hello from SMPPSim Hello from SMPPSim
Hello from SMPPSim Hello from SMPPSim Hello from SMPPSim]]></MESSAGE>
<MESSAGE-IDENTIFIER><![CDATA[111]]></MESSAGE-IDENTIFIER>
<MESSAGE-PARTS><![CDATA[2]]></MESSAGE-PARTS>
<MESSAGE-PART-NUMBER><![CDATA[1]]></MESSAGE-PART-NUMBER>
</XML>
```

4.1.4.2 Delivery Receipt [Delivery connector url]

```
<REQ>
<FEATURE>DELIVERY-RECEIPT</FEATURE>
<TIME-STAMP>{0}</TIME-STAMP>
<TRANSACTION-ID>{1}</TRANSACTION-ID>
<SMPP-ID>{2}</SMPP-ID>
<COMMAND-ID>{3}</COMMAND-ID>
<ESM-CLASS>{4}</ESM-CLASS>
<OA>{5}</OA>
<DA>{6}</DA>
<DCS>{7}</DCS>
<SMS>{8}</SMS>
<MESSAGE-ID>{10}</MESSAGE-ID>
```

```
</REQ>
```

Sample

```
<REQ>
<FEATURE>DELIVERY-RECEIPT</FEATURE>
<TIME-STAMP><![CDATA[20210730194217]]></TIME-STAMP>
<TRANSACTION-ID><![CDATA[74378]]></TRANSACTION-ID>
<SMPP-ID><![CDATA[airtel]]></SMPP-ID>
<COMMAND-ID><![CDATA[5]]></COMMAND-ID>
<OA><![CDATA[222222222]]></OA>
<DA><![CDATA[8688492301]]></DA>
<DCS><![CDATA[0]]></DCS>
<SMS><![CDATA[id:1 sub:001 dlvrd:001 submit date:2107301942 done
date:2107301942 stat:DELIVRD err:000 Text:I called You at Wed,]]></SMS>
<MESSAGE-ID><![CDATA[1]]></MESSAGE-ID>
</REQ>
```

4.2 ESME Json Request Packet

Adapters: http://localhost:8081/ESME/HttpJsonAdapter

4.2.1 Submit_sm

```
{
  "featureId":"submit_sm",
  "timeStamp":25062009103510,
  "respUrl":"http://localhost:8080/ESME/TestServlet",
  "parameters":{
    "reqTransactionId":0,
    "smscId":"airtel",
    "submitSm":{
      "sourceAddrTon": "0",
      "sourceAddrNpi": "0",
      "sourceAddr":1234,
      "destAddrTon": "0",
      "destAddrNpi": "0",
      "destinationAddr":11111111,
      "esmClass": "0",
      "concatenatedSMesmClass": "0",
      "dataCoding":0,
      "shortMessage":"UkNIUiAyMjIy"
    }
  }
}
```

Instant response

```
{
   "Response": {
      "timeStamp": "23072021104848",
        "status": "0"
    }
}
```

Final response

```
{
   "status":"0",
   "statusCode":"0",
   "timeStamp":"23072021122642",
   "parameters":{
      "reqTransactionId":"0",
      "respTransactionId":"0",
      "smscId":"airtel"
   }
}
```

4.2.2 Synch_Submit_sm

```
{
   "featureId": "synch_submit_sm",
   "timeStamp": "09092015184542",
   "parameters": {
      "reqTransactionId": "1",
      "smscId": "airtel",
      "submitSm": {
         "sourceAddrTon": "0",
         "sourceAddrNpi": "0",
         "sourceAddr": "8688492301",
         "destAddrTon": "0",
         "destAddrNpi": "0",
         "destinationAddr": "222222222",
         "esmClass": "0",
         "concatenatedSMesmClass": "0",
         "dataCoding":0,
         "shortMessage":
"SSBjYWxsZWQgWW91IGF0IFdlZCwgOSBTZXAgMjAxNSAxODoxMDoyNQ==",
         "registeredDelivery": "1"
      }
   }
}
```

Synch_submit_response

```
{
    "status": "0",
    "statusCode": "0",
    "timeStamp": "23072021123007",
    "parameters": {
        "reqTransactionId": "2",
        "respTransactionId": "1",
        "smscId": "airtel",
        "messageId": "38"
    }
}
```

4.2.3 MESSAGE PAYLOAD & UDH

4.2.3.1 UDH Support Support [Primary Method]

```
{
  "featureId": "synch_submit_sm",
  "timeStamp": "09092015184542",
  "parameters": {
    "reqTransactionId": "1",
    "smscId": "airtel",
    "submitSm": {
      "sourceAddrTon": "0",
      "sourceAddrNpi": "0",
      "sourceAddr": "8688492301",
      "destAddrTon": "0",
      "destAddrNpi": "0",
      "destinationAddr": "222222222",
      "esmClass": "64",
      "concatenatedSMesmClass": "64",
      "shortMessage":
"SSBjYWxsZWQgWW91IGF0IFdlZCwgOSBTZXAgMjAxNSAxODoxMDoyNQ==",
      "registeredDelivery": "1"
    }
  }
}
```

4.2.3.2 Message Payload [SMSC Support Required]

```
{
  "featureId": "synch_submit_sm",
  "timeStamp": "09092015184542",
  "parameters": {
    "reqTransactionId": "1",
    "smscId": "airtel",
    "submitSm": {
      "sourceAddrTon": "0",
      "sourceAddrNpi": "0",
      "sourceAddr": "8688492301",
      "destAddrTon": "0",
      "destAddrNpi": "0",
      "destinationAddr": "222222222",
      "esmClass": "0",
      "concatenatedSMesmClass": "0",
      "shortMessage":
"SSBjYWxsZWQgWW91IGF0IFdlZCwgOSBTZXAgMjAxNSAxODoxMDoyNQ==",
      "messagePayload":"true",
      "registeredDelivery": "1"
    }
  }
}
```

4.2.4ESME MO & Delivery Packets [in smpp.xml]

4.2.4.1 MO Flow [Connector url]

```
{
   "timeStamp":"{0}",
   "transactionId":"{1}",
   "smscID":"{2}",
   "esmClass":"{4}",
   "orginatingAddress":"{5}",
   "destinationAddress":"{6}",
   "dcs":"{7}",
   "sms":"{8}",
   "byteSms":"{9}",
   "optionalParamText":"{11}",
   "message":"{17}",
   "messageIdentifier":"{18}",
   "messageParts":"{19}",
   "messagePartNumber":"{20}"
}
```

Sample

```
{
"timeStamp": "20210723191615",
"transactionId": "38592",
"smscID": "airtel",
"esmClass": "64",
"orginatingAddress": "3333333333",
"destinationAddress": "337788665522",
"dcs": "0",
"sms": " Hello from SMPPSim   Hello from SMPPSim    Hello from SMPPSim   Hello
from SMPPSim   Hello from SMPPSim   Hello from SMPPSim   Hello from SMPPSim
Hello from SMPPSim   Hello from SMPPSim ",
"byteSms":
"IEhlbGxvIGZyb20gU01QUFNpbSAgIEhlbGxvIGZyb20gU01QUFNpbSAgICBIZWxs
byBmcm9tIFNN
UFBTaW0gICBIZWxsbyBmcm9tIFNNUFBTaW0gICBIZWxsbyBmcm9tIFNNUFBTa
W0gICBIZWxsbyBm
cm9tIFNNUFBTaW0gICBIZWxsbyBmcm9tIFNNUFBTaW0gICBIZWxsbyBmcm9tIFN
NUFBTaW0gICBI
ZWxsbyBmcm9tIFNNUFBTaW0gIA==",
"optionalParamText":"",
"message":" from SMPPSim   Hello from SMPPSim    Hello from SMPPSim   Hello
from SMPPSim   Hello from SMPPSim   Hello from SMPPSim   Hello from SMPPSim
Hello from SMPPSim   Hello from SMPPSim  ",
"messageIdentifier":"111",
"messageParts":"2",
"messagePartNumber":"1"
}
```

4.2.4.2 Delivery Receipt [Delivery connector url]

```
{
   "featureId":"DELIVERY-RECEIPT",
   "timeStamp":"{0}",
   "transactionId":"{1}",
   "smscID":"{2}",
   "commandId":"{3}",
   "orginatingAddress":"{5}",
   "destinationAddress":"{6}",
   "dcs":"{7}",
   "sms":"{8}",
   "messageId":"{10}"
}
```

Sample

```
{"featureId": "DELIVERY-RECEIPT",
"timeStamp": "20210723191555",
"transactionId": "38591",
"smscID": "airtel",
"commandId": "5",
"orginatingAddress": "222222222",
"destinationAddress": "8688492301",
"dcs": "2",
"sms": "id:2 sub:001 dlvrd:001 submit date:2107231915 done date:2107231915
stat:DELIVRD err:000 Text:I called You at Wed",
"messageId": "2"
}
```

5 Definitions

Terms in json	Terms in xml	Size (octets) & Type	Description
featureId	FEATURE	string	Feature of the esme
timeStamp	TIME-STAMP		Time stamp
reqTransactionId	REQ-TRANSACTION-ID	integer	Transaction id of the request
respUrl	RESPONSE-URL	string	Url to which response of submit_sm feature should send, not needed in synch_submit_sm
submitSm	SUBMIT-SM		
smscId	SMSC-ID	string	Unique id that is used to identify the specific smsc.
sourceAddr	SOURCE-ADDR	Var. max 21, C Octet String	Address of SME which originated this message.
sourceAddrTon	SOURCE-ADDR-TON	1, Integer	Type of Number for source address. If not known, set to NULL (Unknown).
sourceAddrNpi	SOURCE-ADDR-NPI	1, Integer	Numbering Plan Indicator for source address. If not known, set to NULL (Unknown).
destinationAddr	DESTINATION-ADDR	Var. max 21, C Octet String	Destination address of this short message. For mobile terminated messages, this is the directory number of the recipient MS.
destAddrTon	DEST-ADDR-TON	1, Integer	Type of Number for destination.
destAddrNpi	DEST-ADDR-NPI	1, Integer	Numbering Plan Indicator for destination.
esmClass	ESM-CLASS	1, Integer	Indicates Message Mode & Message Type. 0→ short message 64→ long message[UDHI flag]
concatenatedSMesmClass	concatenatedSMesmClass	1, Integer	Indicates Message Mode & Message Type. 0→ short message

Terms in json	Terms in xml	Size (octets) & Type	Description
			64→ long message[UDHI flag]
dataCoding	DATA-CODING	1, Integer	Defines the encoding scheme of the short message user data. 0 → English 8 → Other Language
shortMessage	SHORT-MESSAGE	Var, 0-254,Octet String	Up to 254 octets of short message user data. The exact physical limit for short_message size may vary according to the underlying network. Applications which need to send messages longer than 254 octets should use the message_payload parameter.
messagePayload	MESSAGE-PAYLOAD	boolean	True → sends long message in a single packet, without multi-part False → sends long message in multi-part, with UDH
registeredDelivery	REGISTERED-DELIVERY	1, Integer	Indicator to signify if an SMSC delivery receipt or an SME acknowledgement is required. Possible values are 0 & 1. 0 → no delivery receipt 1 → delivery receipt
protocolId	PROTOCOL-ID	1, Integer	Protocol Identifier. 64 → silent 0 → normal
priorityFlag	PRIORITY-FLAG	1, Integer	Designates the priority level of the message.
replaceIfPresentFlag	REPLACE-IF-PRESENT-FLAG	1, Integer	Flag indicating if submitted message should replace an existing message.
smDefaultMsgId	SM-DEFAULT-MSG-ID	1, Integer	Indicates the short message to send from a list of pre-defined ('canned') short messages stored on the SMSC. If not using an SMSC canned message, set to NULL.

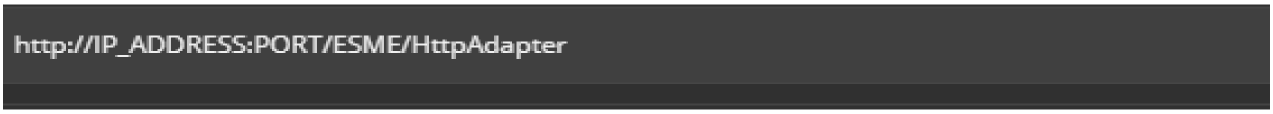
Figure 12 Definitions

Note:	If the values are not present for any of these tags in request xml, it will take the values from the smpp.xml.

6 Request & Response API

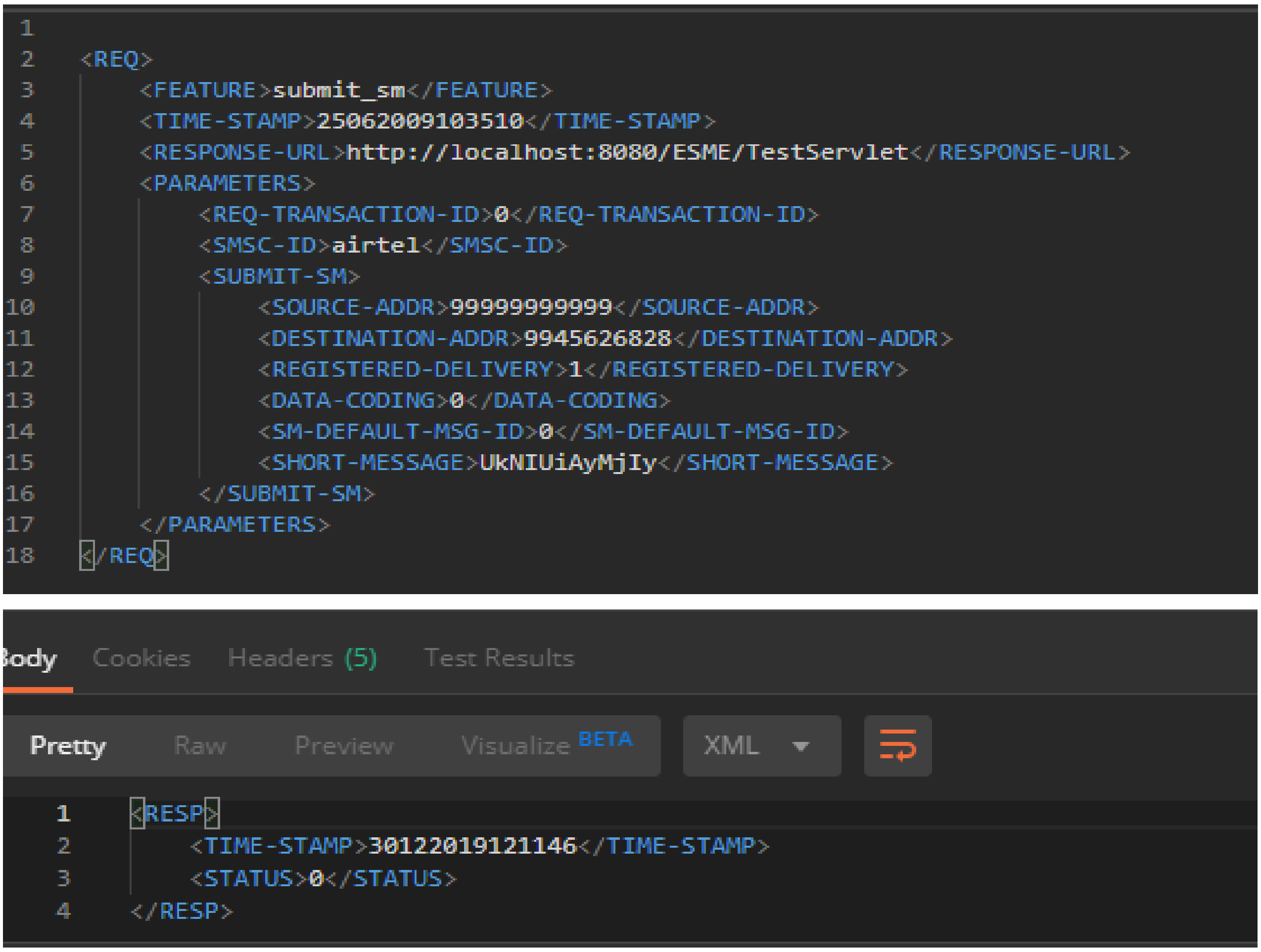
6.1 Submit_SM

Here, in submit_sm, Acknowledgement is given when the request hits the ESME URL and after processing, the original response is given to the *resp URL* attached in the request API. For synch_submit_sm, Request and Response will generate in the same cycle simultaneously, so no acknowledgement is given in that case.



```
http://IP_ADDRESS:PORT/ESME/HttpAdapter
```

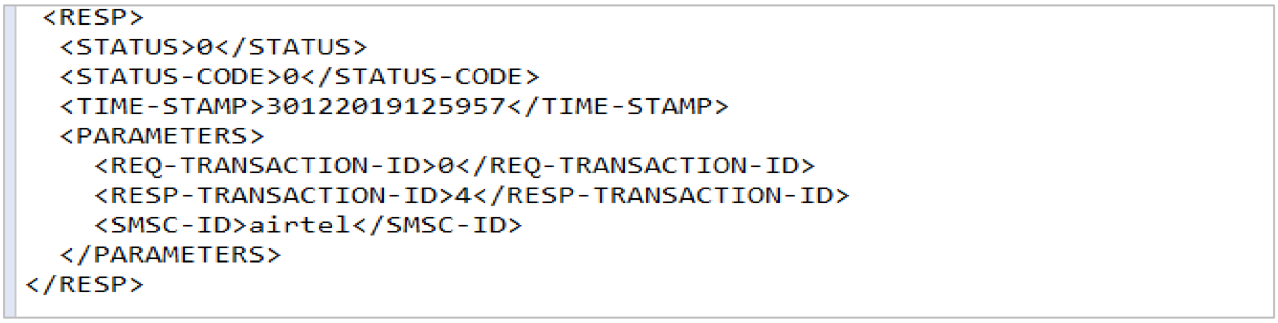
Figure 13 Sample ESME URL



```

<REQ>
    <FEATURE>submit_sm</FEATURE>
    <TIME-STAMP>25062009103510</TIME-STAMP>
    <RESPONSE-URL>http://localhost:8080/ESME/TestServlet</RESPONSE-URL>
    <PARAMETERS>
        <REQ-TRANSACTION-ID>0</REQ-TRANSACTION-ID>
        <SMSC-ID>airtel</SMSC-ID>
        <SUBMIT-SM>
            <SOURCE-ADDR>99999999999</SOURCE-ADDR>
            <DESTINATION-ADDR>9945626828</DESTINATION-ADDR>
            <REGISTERED-DELIVERY>1</REGISTERED-DELIVERY>
            <DATA-CODING>0</DATA-CODING>
            <SM-DEFAULT-MSG-ID>0</SM-DEFAULT-MSG-ID>
            <SHORT-MESSAGE>UkNIUiAyMjIy</SHORT-MESSAGE>
        </SUBMIT-SM>
    </PARAMETERS>
</REQ>
```

```
<RESP>
    <TIME-STAMP>30122019121146</TIME-STAMP>
    <STATUS>0</STATUS>
</RESP>
```

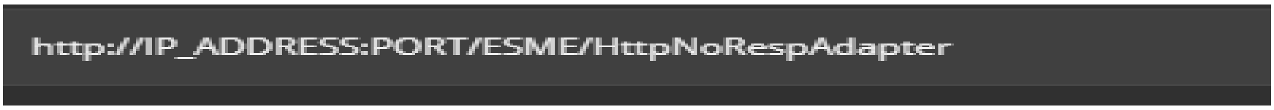
Figure 14 Submit_SM API & Acknowledgement



```
 <RESP>
  <STATUS>0</STATUS>
  <STATUS-CODE>0</STATUS-CODE>
  <TIME-STAMP>30122019125957</TIME-STAMP>
  <PARAMETERS>
    <REQ-TRANSACTION-ID>0</REQ-TRANSACTION-ID>
    <RESP-TRANSACTION-ID>4</RESP-TRANSACTION-ID>
    <SMSC-ID>airtel</SMSC-ID>
  </PARAMETERS>
</RESP>
```

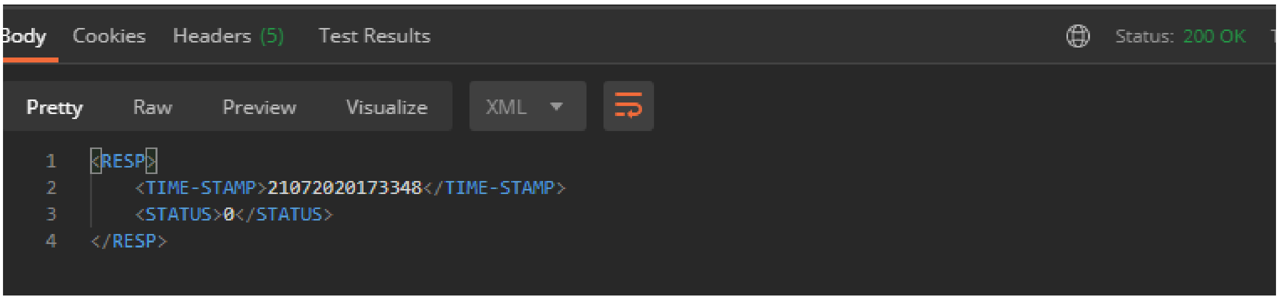
Figure 15 Submit_SM response

If we don’t need the response, then we can use the ***HttpNoRespAdapter*** adapter. It will avoid the extra effort of sending the response to the *resp URL* attached with the request API packet.



```
http://IP_ADDRESS:PORT/ESME/HttpNoRespAdapter
```

Figure 16 Sample ESME URL with HttpNoRespAdapter

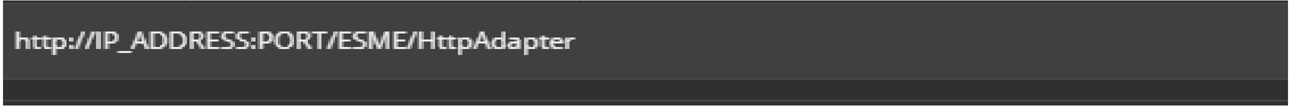


```
<RESP>
    <TIME-STAMP>21072020173348</TIME-STAMP>
    <STATUS>0</STATUS>
</RESP>
```

Figure 17 Acknowledgement received when Submit_SM API is pushed to HttpNoRespAdapter

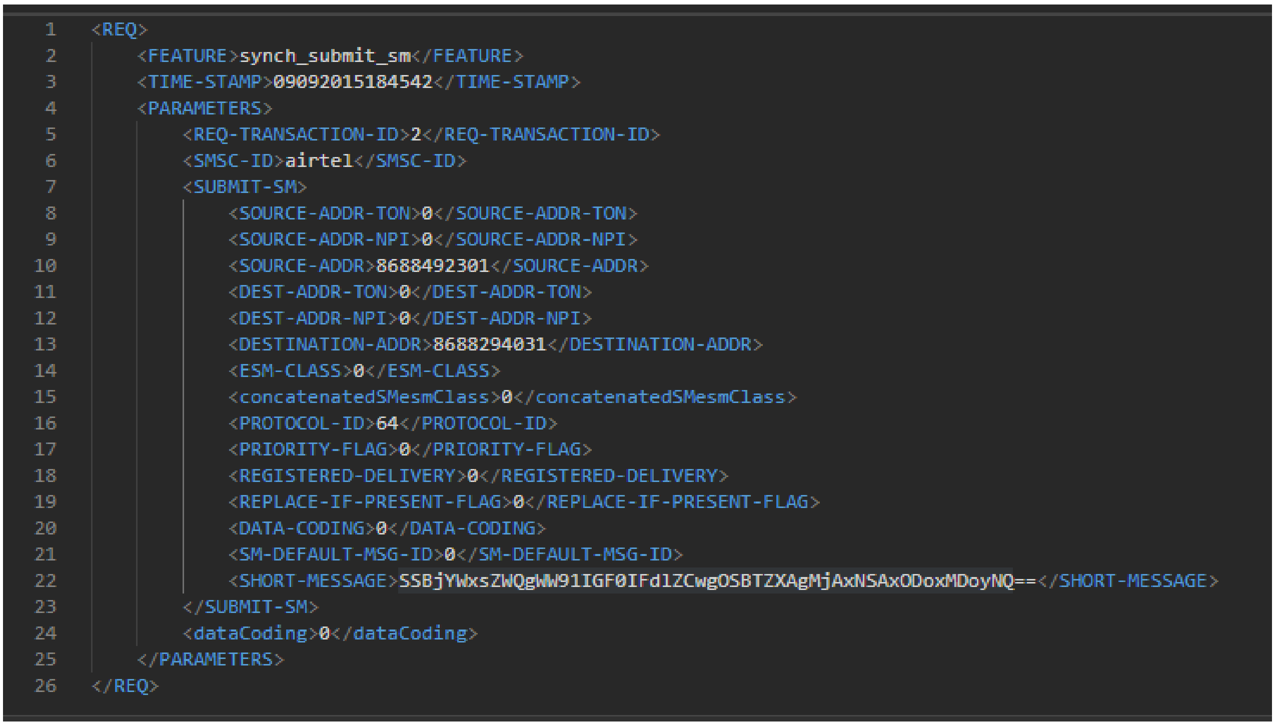
6.2 Synch_Submit_SM

In synch_submit_sm, the response is synchronized with the request. A sample packet of request, response and the adapter of synch_submit_sm is shown below.

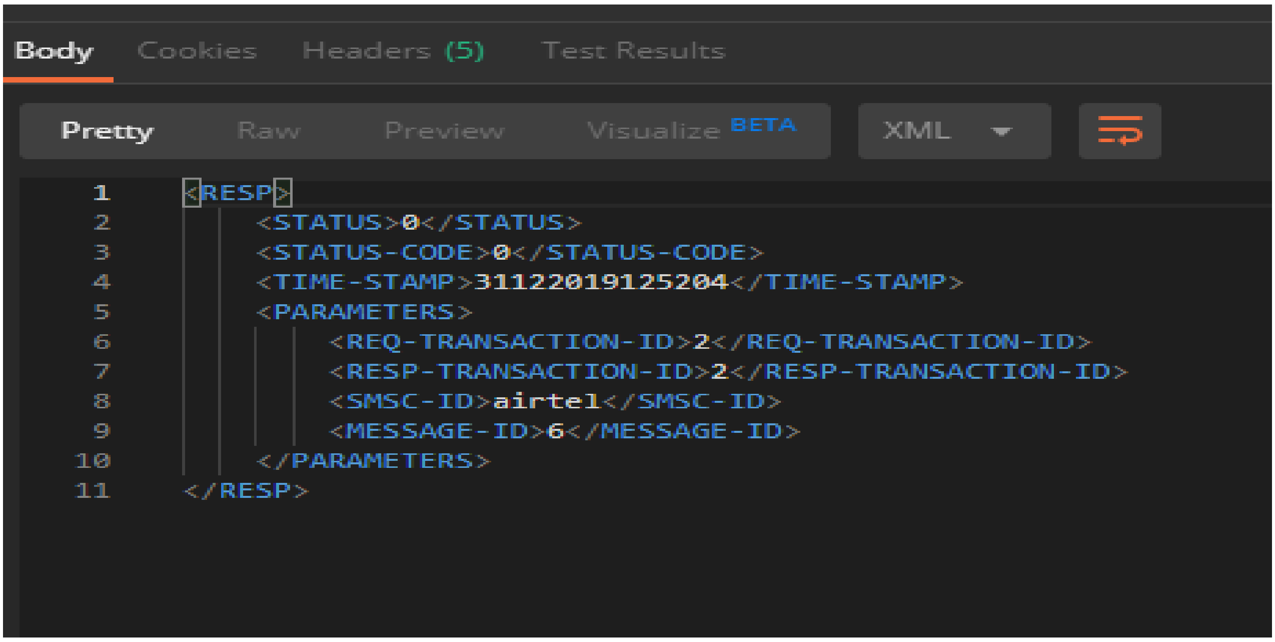


```
http://IP_ADDRESS:PORT/ESME/HttpAdapter
```

Figure 18 Sample ESME URL



```xml
<REQ>
    <FEATURE>synch_submit_sm</FEATURE>
    <TIME-STAMP>09092015184542</TIME-STAMP>
    <PARAMETERS>
        <REQ-TRANSACTION-ID>2</REQ-TRANSACTION-ID>
        <SMSC-ID>airtel</SMSC-ID>
        <SUBMIT-SM>
            <SOURCE-ADDR-TON>0</SOURCE-ADDR-TON>
            <SOURCE-ADDR-NPI>0</SOURCE-ADDR-NPI>
            <SOURCE-ADDR>8688492301</SOURCE-ADDR>
            <DEST-ADDR-TON>0</DEST-ADDR-TON>
            <DEST-ADDR-NPI>0</DEST-ADDR-NPI>
            <DESTINATION-ADDR>8688294031</DESTINATION-ADDR>
            <ESM-CLASS>0</ESM-CLASS>
            <concatenatedSMesmClass>0</concatenatedSMesmClass>
            <PROTOCOL-ID>64</PROTOCOL-ID>
            <PRIORITY-FLAG>0</PRIORITY-FLAG>
            <REGISTERED-DELIVERY>0</REGISTERED-DELIVERY>
            <REPLACE-IF-PRESENT-FLAG>0</REPLACE-IF-PRESENT-FLAG>
            <DATA-CODING>0</DATA-CODING>
            <SM-DEFAULT-MSG-ID>0</SM-DEFAULT-MSG-ID>
            <SHORT-MESSAGE>SSBjYWxsZWQgWW91IGF0IFdlZCwgOSBTZXAgMjAxNSAxODoxMDoyNQ==</SHORT-MESSAGE>
        </SUBMIT-SM>
        <dataCoding>0</dataCoding>
    </PARAMETERS>
</REQ>
```



```xml
<RESP>
    <STATUS>0</STATUS>
    <STATUS-CODE>0</STATUS-CODE>
    <TIME-STAMP>31122019125204</TIME-STAMP>
    <PARAMETERS>
        <REQ-TRANSACTION-ID>2</REQ-TRANSACTION-ID>
        <RESP-TRANSACTION-ID>2</RESP-TRANSACTION-ID>
        <SMSC-ID>airtel</SMSC-ID>
        <MESSAGE-ID>6</MESSAGE-ID>
    </PARAMETERS>
</RESP>
```

Figure 19 Synch_Submit SM API & Response

7 Special Cases

7.1 Message Payload[MT]

A maximum of 254 octets (according to standard smpp configuration) can be sent in submit_sm. To send larger user data sizes [from ESME to SMSC], there are two methods.

7.1.1Multipart message with UDH

MO-Router needs to send splitted message segments with additional message header parameters (**UDH - User Data Header**) . When sending a multipart message, SMPP requires us to send each message part using a separate request. Each request must include a UDH in the message text that indicates that it is part of a multipart message. The two relevant parameters in the PDU body are:

- **esm_class** — set the value to **0x40 (this is 64 in decimal)** to indicate that there is a UDH.
- **short_message** — add the **UDH information** at the beginning of the message.

Message 1 (UDH+153 chars length message)

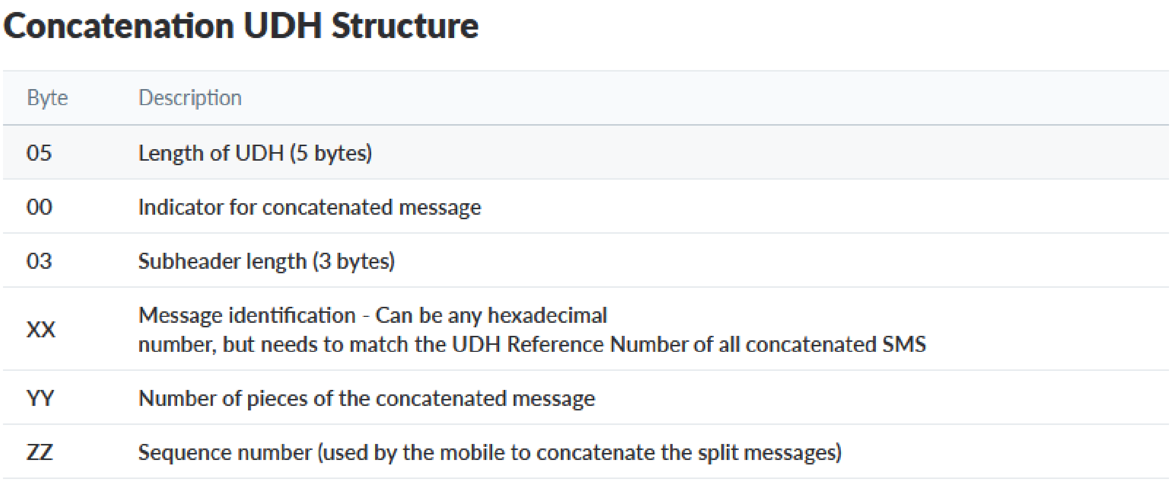
esm_class = 0x40 x1xxxxxx

short_message = **0x05 0x00 0x03 0x05 0x02 0x01** *Barcelona are to appeal against Uefa's proposal to ban Andres Iniesta for an additional Champions League match for allegedly provoking a yellow card. The*

Message 2 (UDH+rest of the message)

esm_class = 0x40 x1xxxxxx

short_message = 0x05 0x00 0x03 0x05 0x02 0x02 *26-year-old Spain midfielder was booked in the quarter-final against Shakhtar Donetsk, meaning he missed the return leg but would then be available.*



Concatenation UDH Structure

Byte	Description
05	Length of UDH (5 bytes)
00	Indicator for concatenated message
03	Subheader length (3 bytes)
XX	Message identification - Can be any hexadecimal number, but needs to match the UDH Reference Number of all concatenated SMS
YY	Number of pieces of the concatenated message
ZZ	Sequence number (used by the mobile to concatenate the split messages)

UDH structure explanation

Maximum Characters Per Concatenated Message

The maximum number of characters per concatenated message depends on the encoding.

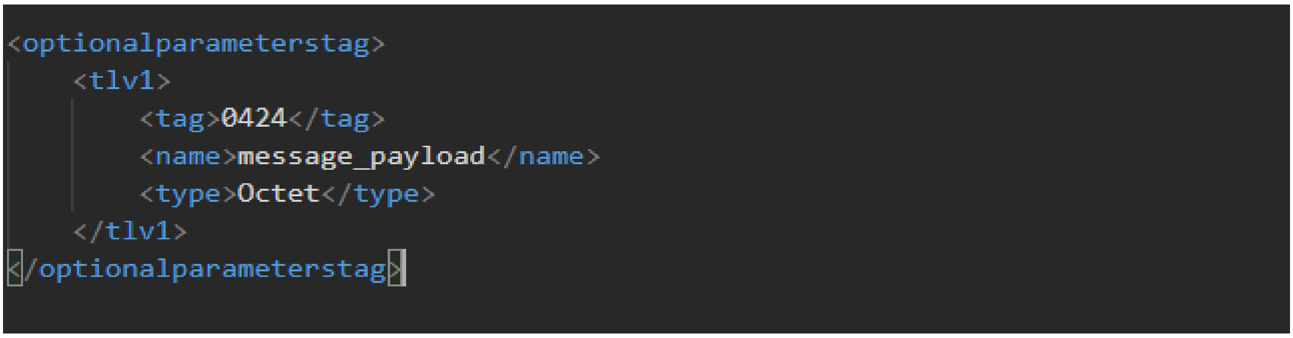
- **153 characters** for 7-bit encoding (e.g., Latin-1/9 and GSM8)
- **134 characters** for 8-bit encoding (Binary)
- **67 characters** for 16-bit encoding (Unicode)

In request xml [Request from MO Router to ESME]

```
<ESM-CLASS>64</ESM-CLASS>
<concatenatedSMesmClass>64</concatenatedSMesmClass>
```

7.1.2Message payload concept

We need to configure the following tags in the **smpp.xml** and add the extra tags in **request xml** for enabling the payload.



```
<optionalparameterstag>
    <tlv1>
        <tag>0424</tag>
        <name>message_payload</name>
        <type>Octet</type>
    </tlv1>
</optionalparameterstag>
```

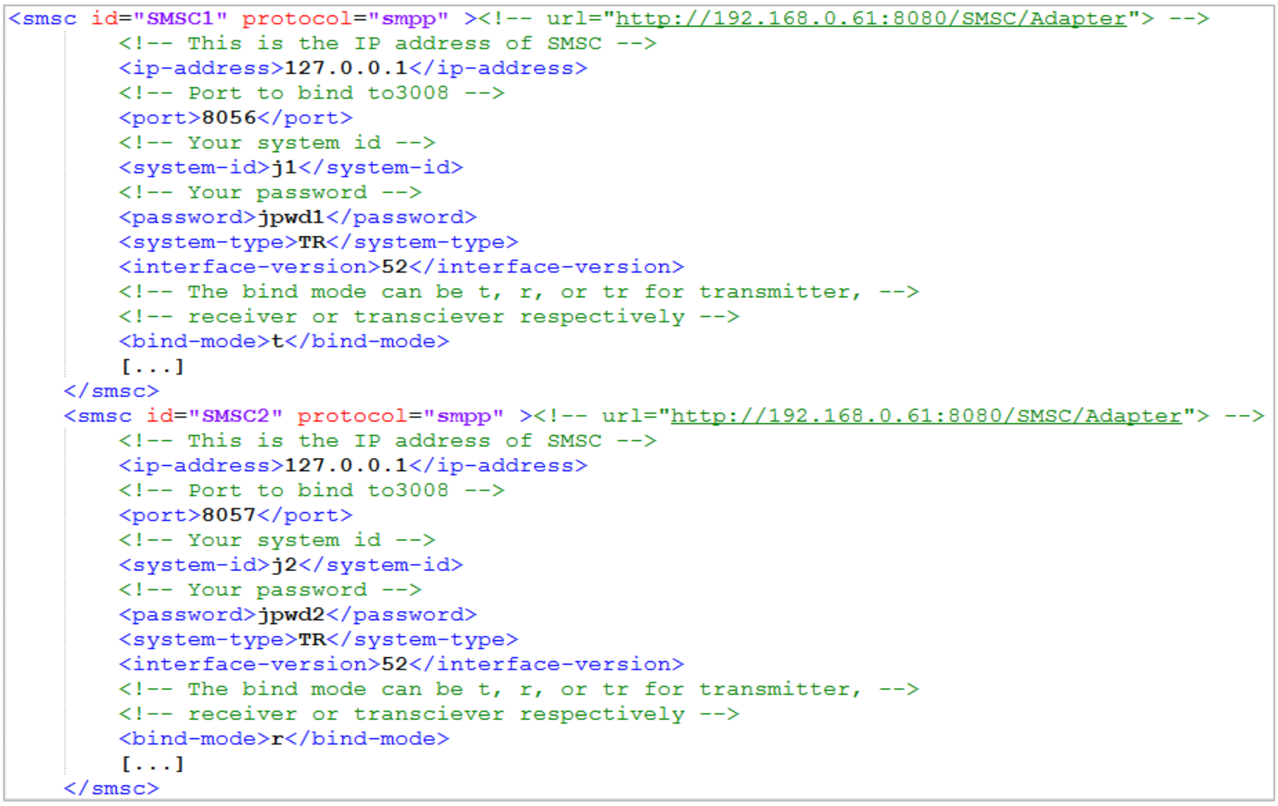
Figure 20 Optional Parameter Tag

In request xml [Request from MO Router to ESME]

```
<ESM-CLASS>0</ESM-CLASS>
<concatenatedSMesmClass>0</concatenatedSMesmClass>
<MESSAGE-PAYLOAD>true</MESSAGE-PAYLOAD>
```

7.2 Multiple SMSC Support

ESME has the multi - smpp support. It can be connected to two or more smsc accounts. We need to configure the accounts separately in the smpp.xml. A ***sample configuration*** is attached here. It will support different bind modes for different smsc accounts.



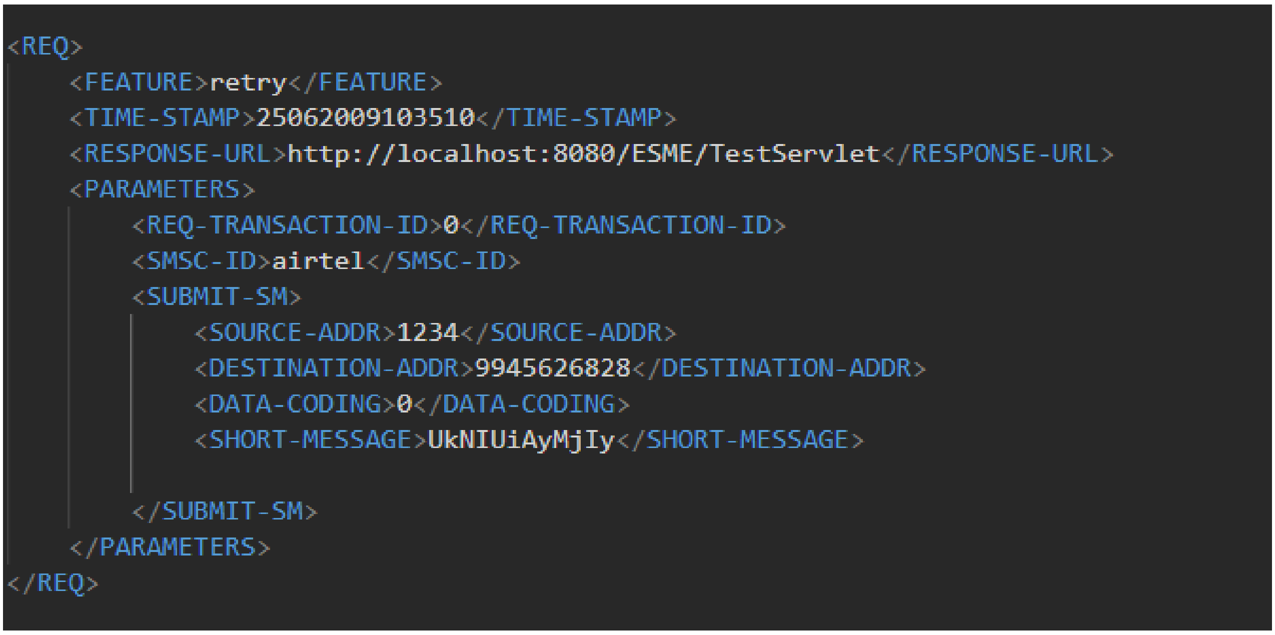
```xml
<smsc id="SMSC1" protocol="smpp" ><!-- url="http://192.168.0.61:8080/SMSC/Adapter"> -->
        <!-- This is the IP address of SMSC -->
        <ip-address>127.0.0.1</ip-address>
        <!-- Port to bind to3008 -->
        <port>8056</port>
        <!-- Your system id -->
        <system-id>j1</system-id>
        <!-- Your password -->
        <password>jpwd1</password>
        <system-type>TR</system-type>
        <interface-version>52</interface-version>
        <!-- The bind mode can be t, r, or tr for transmitter, -->
        <!-- receiver or transciever respectively -->
        <bind-mode>t</bind-mode>
        [...]
    </smsc>
    <smsc id="SMSC2" protocol="smpp" ><!-- url="http://192.168.0.61:8080/SMSC/Adapter"> -->
        <!-- This is the IP address of SMSC -->
        <ip-address>127.0.0.1</ip-address>
        <!-- Port to bind to3008 -->
        <port>8057</port>
        <!-- Your system id -->
        <system-id>j2</system-id>
        <!-- Your password -->
        <password>jpwd2</password>
        <system-type>TR</system-type>
        <interface-version>52</interface-version>
        <!-- The bind mode can be t, r, or tr for transmitter, -->
        <!-- receiver or transciever respectively -->
        <bind-mode>r</bind-mode>
        [...]
    </smsc>
```

Figure 21 Sample Multi - SMSC Support XML Configuration

7.3 Retry

ESME also supports the Retry Feature. In ***Retry Config Properties***, change the value of the synchSubmitSm to retrySubmitSm and make the value of roundRobinSupport to false.

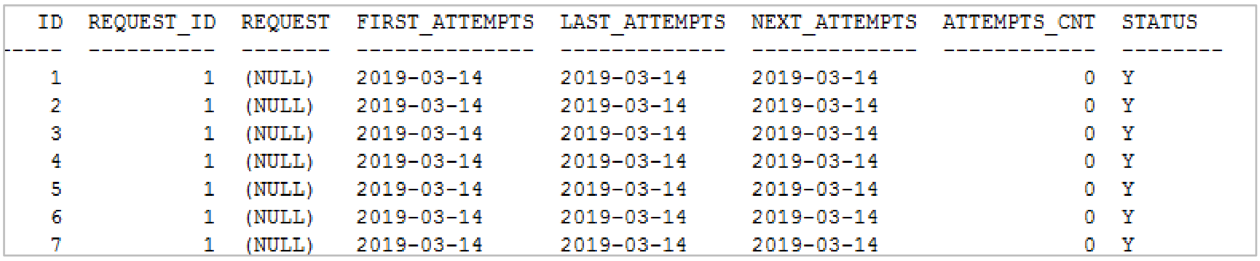
```
synchSubmitSm = retrySubmitSm

roundRobinSupport=false
```



```xml
<REQ>
    <FEATURE>retry</FEATURE>
    <TIME-STAMP>25062009103510</TIME-STAMP>
    <RESPONSE-URL>http://localhost:8080/ESME/TestServlet</RESPONSE-URL>
    <PARAMETERS>
        <REQ-TRANSACTION-ID>0</REQ-TRANSACTION-ID>
        <SMSC-ID>airtel</SMSC-ID>
        <SUBMIT-SM>
            <SOURCE-ADDR>1234</SOURCE-ADDR>
            <DESTINATION-ADDR>9945626828</DESTINATION-ADDR>
            <DATA-CODING>0</DATA-CODING>
            <SHORT-MESSAGE>UkNIUiAyMjIy</SHORT-MESSAGE>

        </SUBMIT-SM>
    </PARAMETERS>
</REQ>
```

Figure 22 Retry XML Configuration

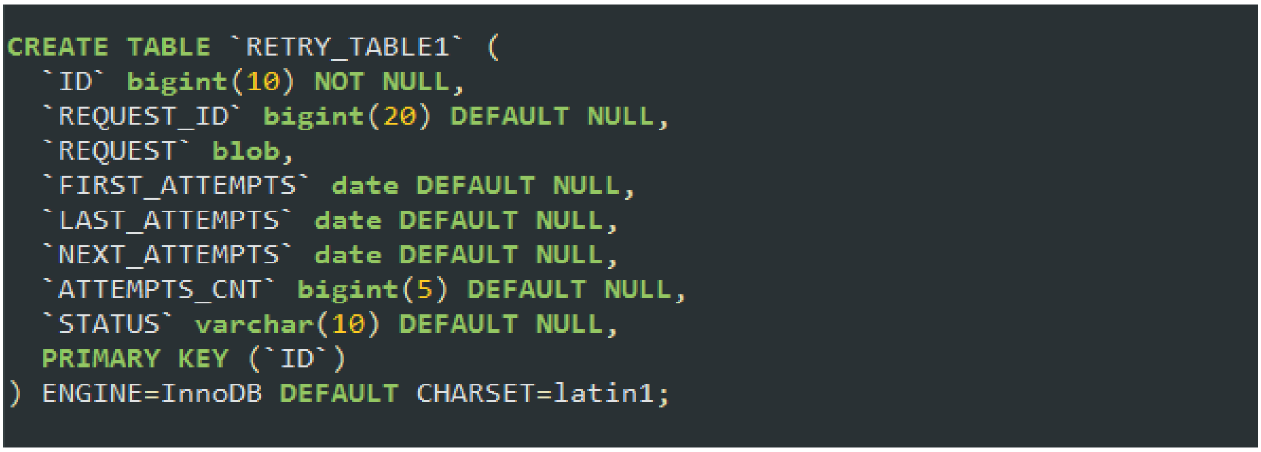
Apart from this, we have a RETRY table configuration consisting of **REQUEST_ID, REQUEST, FIRST_ATTEMPTS, LAST_ATTEMPTS, NEXT_ATTEMPTS, ATTEMPTS_CNT AND STATUS.**



```
    ID  REQUEST_ID  REQUEST  FIRST_ATTEMPTS  LAST_ATTEMPTS  NEXT_ATTEMPTS  ATTEMPTS_CNT  STATUS
-----  ----------  -------  --------------  -------------  -------------  ------------  --------
     1           1  (NULL)   2019-03-14      2019-03-14     2019-03-14                0  Y
     2           1  (NULL)   2019-03-14      2019-03-14     2019-03-14                0  Y
     3           1  (NULL)   2019-03-14      2019-03-14     2019-03-14                0  Y
     4           1  (NULL)   2019-03-14      2019-03-14     2019-03-14                0  Y
     5           1  (NULL)   2019-03-14      2019-03-14     2019-03-14                0  Y
     6           1  (NULL)   2019-03-14      2019-03-14     2019-03-14                0  Y
     7           1  (NULL)   2019-03-14      2019-03-14     2019-03-14                0  Y
```

Figure 23 Retry Table

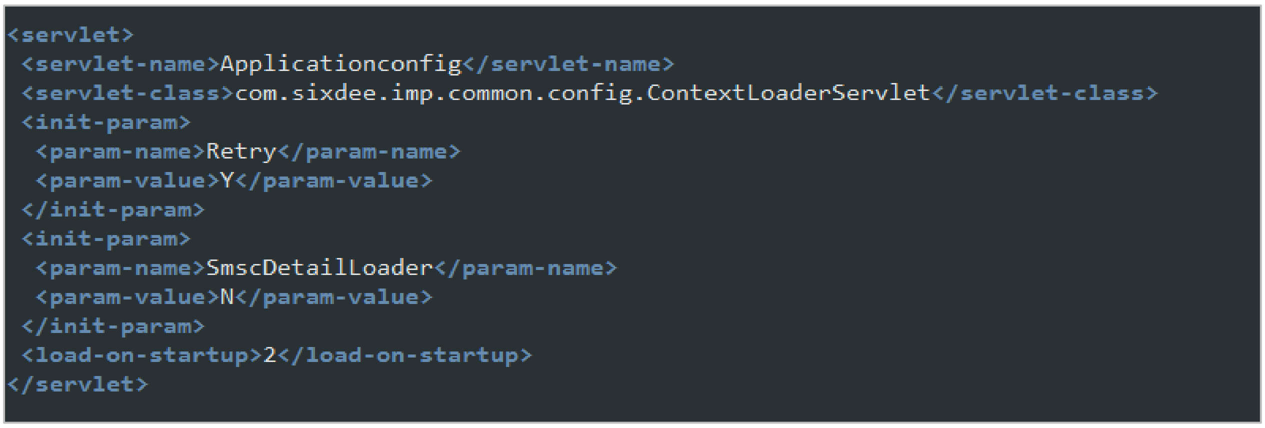
REQUEST - request object	FIRST_ATTEMPTS - the date of the first attempt
ATTEMPTS_CNT - the count of attempts	LAST_ATTEMPTS - the date of the last attempt
STATUS - Y or N	NEXT_ATTEMPTS - the date of the next attempt



```
CREATE TABLE `RETRY_TABLE1` (
  `ID` bigint(10) NOT NULL,
  `REQUEST_ID` bigint(20) DEFAULT NULL,
  `REQUEST` blob,
  `FIRST_ATTEMPTS` date DEFAULT NULL,
  `LAST_ATTEMPTS` date DEFAULT NULL,
  `NEXT_ATTEMPTS` date DEFAULT NULL,
  `ATTEMPTS_CNT` bigint(5) DEFAULT NULL,
  `STATUS` varchar(10) DEFAULT NULL,
  PRIMARY KEY (`ID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Figure 24 Retry Table Database Schema

To enable Retry, we need to change the init-param value of Retry inside the web.xml. The web.xml will be available on the $WILDFLY_HOME/standalone/deployments/ Esme.war. The param-value, which is default N, should be changed into Y to enable the retry feature. Then follow these ***retry_configurations***.

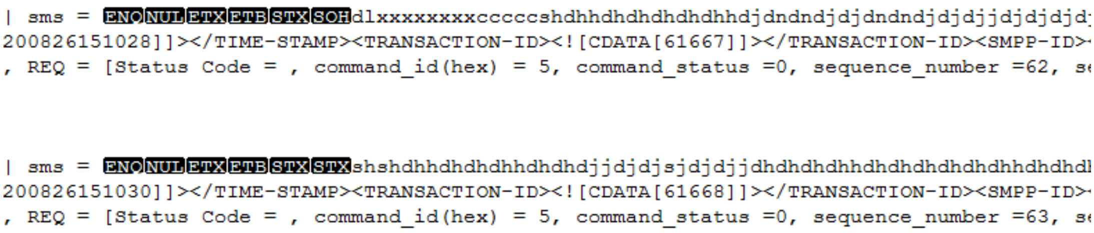


```
<servlet>
 <servlet-name>Applicationconfig</servlet-name>
 <servlet-class>com.sixdee.imp.common.config.ContextLoaderServlet</servlet-class>
 <init-param>
  <param-name>Retry</param-name>
  <param-value>Y</param-value>
 </init-param>
 <init-param>
  <param-name>SmscDetailLoader</param-name>
  <param-value>N</param-value>
 </init-param>
 <load-on-startup>2</load-on-startup>
</servlet>
```

Figure 25 Web.XML Configurations

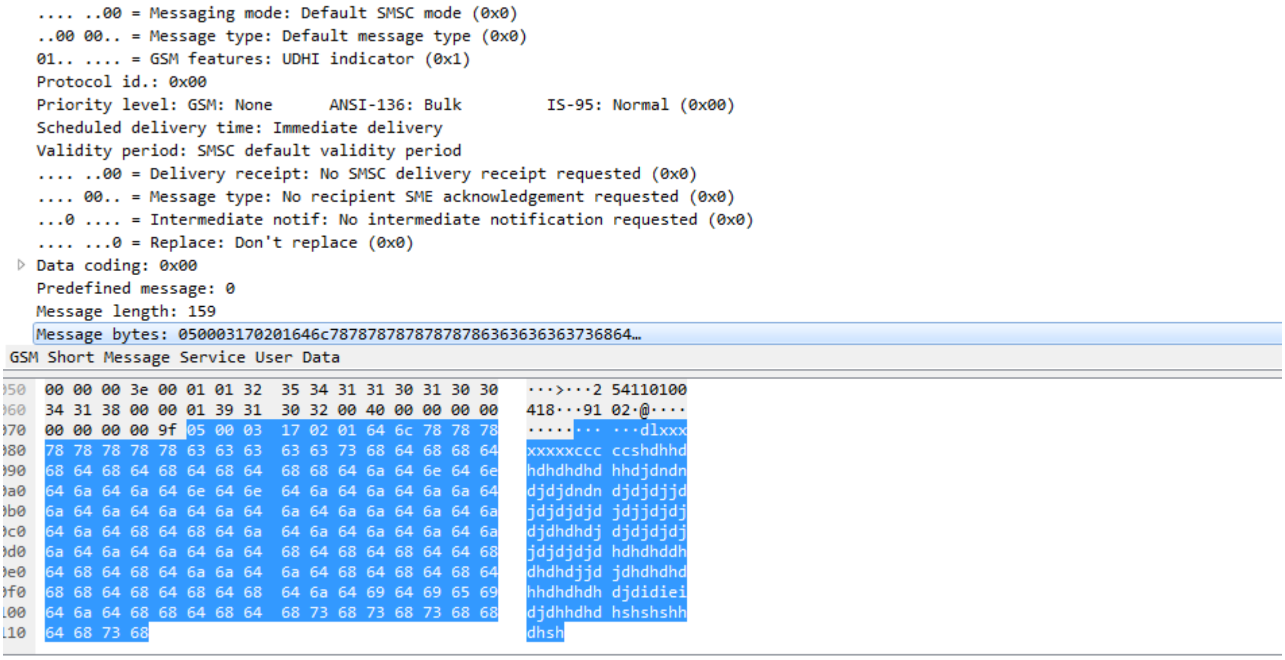
7.4 ESME RECIEVEING ABOVE 160 chars[MO]

SMSC usually transfers characters less than 160 in the message bytes to the ESME (MO flow). But **sometimes, SMSC will transfer above 160 characters** to ESME. And this long message comes as multiple messages i.e. **part-messages** to ESME with corresponding **UDH headers.** The first few characters of each part message are reserved for these UDH headers, which may **appear as junk characters** in the trace and logs. These junk characters a.k.a UDH headers are needed for **Mo Router** or any other equivalent module **to do the process of message concatenation.** Mo router **saves each part message and concatenate** them using their own internal logic with the help of these UDH headers.



```
| sms = ENQNULETXETBSTXSOHdlxxxxxxxxcccccshdhhdhdhdhdhdhhdjdndndjdjdndndjdjdjjdjdjdjd
200826151028]]></TIME-STAMP><TRANSACTION-ID><![CDATA[61667]]></TRANSACTION-ID><SMPP-ID>
, REQ = [Status Code = , command_id(hex) = 5, command_status =0, sequence_number =62, s


| sms = ENQNULETXETBSTXSTXshshdhhdhdhdhhdhdhdjjdjdjsjdjdjjdhdhdhdhhdhdhdhdhdhdhhdhdhdl
200826151030]]></TIME-STAMP><TRANSACTION-ID><![CDATA[61668]]></TRANSACTION-ID><SMPP-ID>
, REQ = [Status Code = , command_id(hex) = 5, command_status =0, sequence_number =63, s
```

Figure 26: UDH headers in the message is being seen as junk



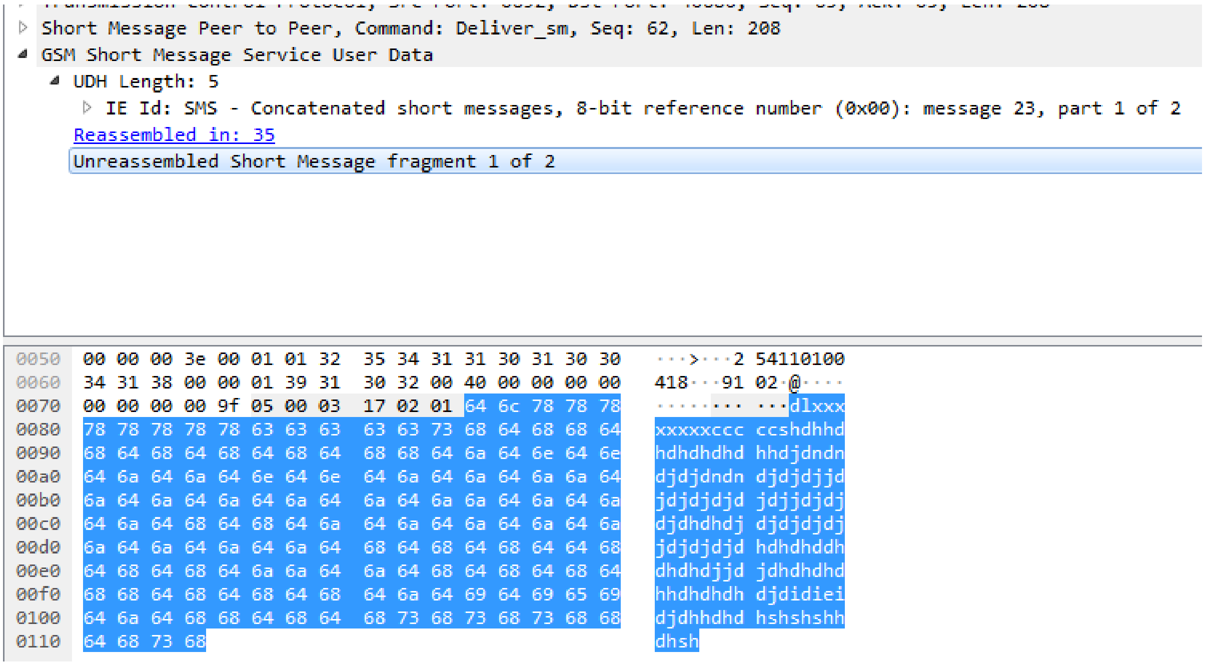
```
.... ..00 = Messaging mode: Default SMSC mode (0x0)
..00 00.. = Message type: Default message type (0x0)
01.. .... = GSM features: UDHI indicator (0x1)
Protocol id.: 0x00
Priority level: GSM: None      ANSI-136: Bulk        IS-95: Normal (0x00)
Scheduled delivery time: Immediate delivery
Validity period: SMSC default validity period
.... ..00 = Delivery receipt: No SMSC delivery receipt requested (0x0)
.... 00.. = Message type: No recipient SME acknowledgement requested (0x0)
...0 .... = Intermediate notif: No intermediate notification requested (0x0)
.... ...0 = Replace: Don't replace (0x0)
Data coding: 0x00
Predefined message: 0
Message length: 159
Message bytes: 050003170201646c78787878787878786363636363736864…
GSM Short Message Service User Data

050  00 00 00 3e 00 01 01 32  35 34 31 31 30 31 30 30   ···>···2 54110100
060  34 31 38 00 00 01 39 31  30 32 00 40 00 00 00 00   418···91 02·@····
070  00 00 00 00 9f 05 00 03  17 02 01 64 6c 78 78 78   ········ ···dlxxx
080  78 78 78 78 78 63 63 63  63 63 73 68 64 68 68 64   xxxxxccc ccshdhhd
090  68 64 68 64 68 64 68 64  68 68 64 6a 64 6e 64 6e   hdhdhdhd hhdjdndn
0a0  64 6a 64 6a 64 6e 64 6e  64 6a 64 6a 64 6a 6a 64   djdjdndn djdjdjjd
0b0  6a 64 6a 64 6a 64 6a 64  6a 64 6a 6a 64 6a 64 6a   jdjdjdjd jdjjdjdj
0c0  64 6a 64 68 64 68 64 6a  64 6a 64 6a 64 6a 64 6a   djdhdhdj djdjdjdj
0d0  6a 64 6a 64 6a 64 6a 64  68 64 68 64 68 64 64 68   jdjdjdjd hdhdhddh
0e0  64 68 64 68 64 6a 6a 64  6a 64 68 64 68 64 68 64   dhdhdjjd jdhdhdhd
0f0  68 68 64 68 64 68 64 68  64 6a 64 69 64 69 65 69   hhdhdhdh djdidiei
100  64 6a 64 68 68 64 68 64  68 73 68 73 68 73 68 68   djdhhdhd hshshshh
110  64 68 73 68                                        dhsh
```

Figure 27: UDH headers coming as dots before message in trace

The **<SMS-BYTE>** tag in the **request xml to MO router** [refer connector URL section] from ESME gives the incoming message from SMSC in **byte format**, but **base 64 encoded.** The same can be decoded into bytes, by MO Router, **for getting the UDH headers** in order to use them for the operation of concatenation. [MOR internal logic]

Note: **ESME don't concatenate the message using UDH.**

The following images can be used for the better understanding of the concept in detail.



```
▷ Short Message Peer to Peer, Command: Deliver_sm, Seq: 62, Len: 208
◢ GSM Short Message Service User Data
  ◢ UDH Length: 5
    ▷ IE Id: SMS - Concatenated short messages, 8-bit reference number (0x00): message 23, part 1 of 2
    Reassembled in: 35
    Unreassembled Short Message fragment 1 of 2

0050  00 00 00 3e 00 01 01 32  35 34 31 31 30 31 30 30   ···>···2 54110100
0060  34 31 38 00 00 01 39 31  30 32 00 40 00 00 00 00   418···91 02·@····
0070  00 00 00 00 9f 05 00 03  17 02 01 64 6c 78 78 78   ········ ···dlxxx
0080  78 78 78 78 78 63 63 63  63 63 73 68 64 68 68 64   xxxxxccc ccshdhhd
0090  68 64 68 64 68 64 68 64  68 68 64 6a 64 6e 64 6e   hdhdhdhd hhdjdndn
00a0  64 6a 64 6a 64 6e 64 6e  64 6a 64 6a 64 6a 6a 64   djdjdndn djdjdjjd
00b0  6a 64 6a 64 6a 64 6a 64  6a 64 6a 6a 64 6a 64 6a   jdjdjdjd jdjjdjdj
00c0  64 6a 64 68 64 68 64 6a  64 6a 64 6a 64 6a 64 6a   djdhdhdj djdjdjdj
00d0  6a 64 6a 64 6a 64 6a 64  68 64 68 64 68 64 64 68   jdjdjdjd hdhdhddh
00e0  64 68 64 68 64 6a 6a 64  6a 64 68 64 68 64 68 64   dhdhdjjd jdhdhdhd
00f0  68 68 64 68 64 68 64 68  64 6a 64 69 64 69 65 69   hhdhdhdh djdidiei
0100  64 6a 64 68 68 64 68 64  68 73 68 73 68 73 68 68   djdhhdhd hshshshh
0110  64 68 73 68                                         dhsh
```

Fig 28: UDH details from the trace showing that it's the first part-message out of 2

<SMS>
<![CDATA[␅�␃␗␂␁
dlxxxxxxxxcccccshdhhdhdhdhdhdhhdjdndndjdjdndndjdjdjjdjdjdjdjdjjdjdjdjdhdhdjdjdjdjdjjdjdjdjdhdhdhddhdhdhdjjdjdhdhdhdhhdhdhdhdjdidieidjdhhdhdhshshshhdhsh]]>
</SMS>

<SMS-BYTE>
<![CDATA[BQADFwIBZGx4eHh4eHh4eGNjY2Njc2hkaGhkaGRoZGhkaGRoaGRqZG5kbmRqZGpkbmRuZGpkamRqamRqZGpkamRqZGpkampkamRqZGpkaGRoZGpkamRqZGpkampkamRqZGpkaGRoZGhkZGhkaGRoZGpqZGpkaGRoZGhkaGhkaGRoZGhkamRpZGllaWRqZGhoZGhkaHNoc2hzaGhkaHNo]]>
</SMS-BYTE>
</XML>

Fig 29: SMS-BYTE & SMS tag from connector URL [request xml to MO-Router]

Later, on request of MOR team, we devised a solution to **decode the above UDH headers** from the **<SMS_BYTE>** and sent them as separate xml tags.

<messageformat><![CDATA[<XML><TIME-STAMP>{0}</TIME-STAMP><TRANSACTION-ID>{1}</TRANSACTION-ID><SMPP-ID>{2}</SMPP-ID><ESM-CLASS>{4}</ESM-CLASS><OA>{5}</OA><DA>{6}</DA><DCS>{7}</DCS><SMS>{8}</SMS><SMS-BYTE>{9}</SMS-BYTE><OPTIONAL-PARAM>{11}</OPTIONAL-PARAM><MESSAGE>{17}</MESSAGE><MESSAGE-IDENTIFIER>{18}</MESSAGE-IDENTIFIER><MESSAGE-PARTS>{19}</MESSAGE-PARTS><MESSAGE-PART-NUMBER>{20}</MESSAGE-PART-NUMBER></XML>]]></messageformat>

Refer the connector xml section in **SMPP.xml** to see what these numbers are meant.

7.5 ELK logging

ESME supports the ELK logging. ELK stands for Elasticsearch, Logstash, and Kibana. In the ELK stack, Logstash extracts the logging data or other events from different input sources. It processes the events and later stores it in Elasticsearch. Kibana is a web interface, which accesses the logging data form Elasticsearch and visualizes it.

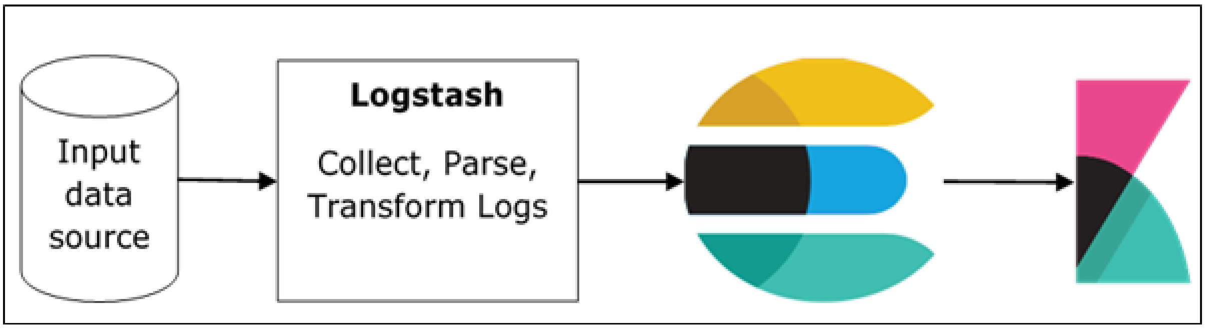


Fig 30: ELK working

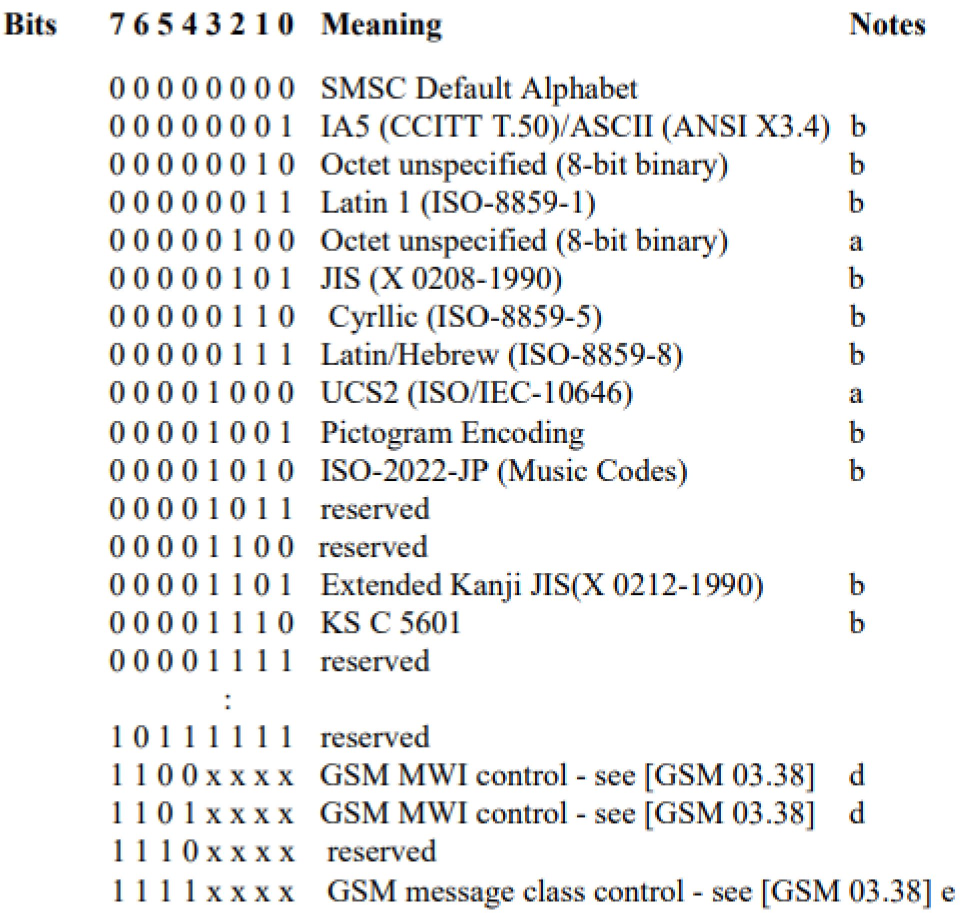
For sending logs into ELK, we added new configuration, <logs> in the **Smpp.xml**

```
<logs>
<formatPattern><![CDATA[TimeStamp::{1,date,yyyy-MM-dd'T'HH:mm:ss XXX}||ClientTransactionId::{3}||MSISDN::{15}||TransactionId::{2}||OfferCode::0||TransactionDate::{4}||statusCode::{5}||Feature::{6}||SmscId::{0}||commandStatus::{7}||SeqNo::{8}||ServiceType::{9}||SourceAddrTon::{10}||SourceAddrNpi::{11}||DestAddrTon::{13}||DestAddrNpi::{14}||SourceAddr::{12}||EsmClass::{16}||ProtocolId::{17}||RegisteredDelivery::{21}||DataCoding::{23}||ShortMessageHex::{25}||ShortMessage::{32}]]></formatPattern>
</logs>
```

The corresponding placeholders values are described in **ELK_LogEsme_Object_Id_Desc** **.** We can customize the parameters according to the client in <logs> section using the above document.

A new **log4j appender** is also written in the log4j.xml to support ELK Logging.

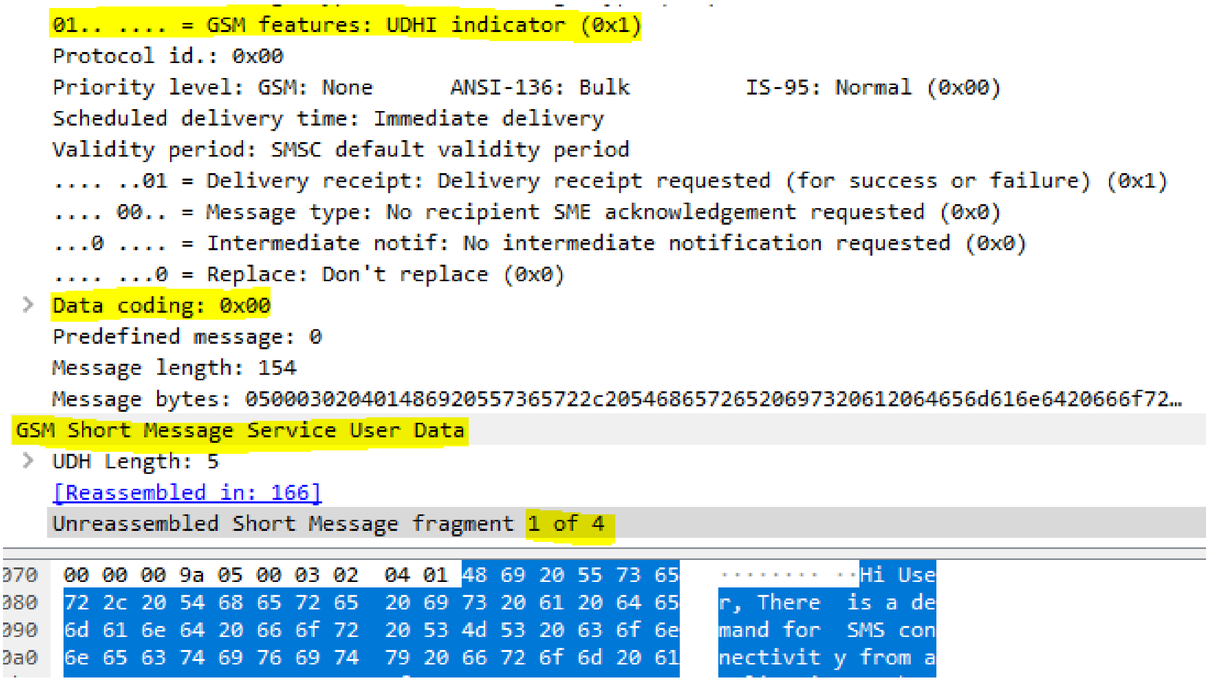
7.6 ESME Data Coding



Bits	7 6 5 4 3 2 1 0	Meaning	Notes
	0 0 0 0 0 0 0 0	SMSC Default Alphabet	
	0 0 0 0 0 0 0 1	IA5 (CCITT T.50)/ASCII (ANSI X3.4)	b
	0 0 0 0 0 0 1 0	Octet unspecified (8-bit binary)	b
	0 0 0 0 0 0 1 1	Latin 1 (ISO-8859-1)	b
	0 0 0 0 0 1 0 0	Octet unspecified (8-bit binary)	a
	0 0 0 0 0 1 0 1	JIS (X 0208-1990)	b
	0 0 0 0 0 1 1 0	Cyrllic (ISO-8859-5)	b
	0 0 0 0 0 1 1 1	Latin/Hebrew (ISO-8859-8)	b
	0 0 0 0 1 0 0 0	UCS2 (ISO/IEC-10646)	a
	0 0 0 0 1 0 0 1	Pictogram Encoding	b
	0 0 0 0 1 0 1 0	ISO-2022-JP (Music Codes)	b
	0 0 0 0 1 0 1 1	reserved	
	0 0 0 0 1 1 0 0	reserved	
	0 0 0 0 1 1 0 1	Extended Kanji JIS(X 0212-1990)	b
	0 0 0 0 1 1 1 0	KS C 5601	b
	0 0 0 0 1 1 1 1	reserved	
	:		
	1 0 1 1 1 1 1 1	reserved	
	1 1 0 0 x x x x	GSM MWI control - see [GSM 03.38]	d
	1 1 0 1 x x x x	GSM MWI control - see [GSM 03.38]	d
	1 1 1 0 x x x x	reserved	
	1 1 1 1 x x x x	GSM message class control - see [GSM 03.38]	e

7.6.1 Data Coding – English

English langauge is the most common usage case. We can use **DCS zero** for english script. **0** → ***SMSC Default Alphabet***. The trace for English language - long multi-part message(above 160 characters) case is available here.



```
    01.. .... = GSM features: UDHI indicator (0x1)
    Protocol id.: 0x00
    Priority level: GSM: None      ANSI-136: Bulk        IS-95: Normal (0x00)
    Scheduled delivery time: Immediate delivery
    Validity period: SMSC default validity period
    .... ..01 = Delivery receipt: Delivery receipt requested (for success or failure) (0x1)
    .... 00.. = Message type: No recipient SME acknowledgement requested (0x0)
    ...0 .... = Intermediate notif: No intermediate notification requested (0x0)
    .... ...0 = Replace: Don't replace (0x0)
  > Data coding: 0x00
    Predefined message: 0
    Message length: 154
    Message bytes: 050003020401486920557365722c20546865726520697320612064656d616e6420666f72…
GSM Short Message Service User Data
  > UDH Length: 5
    [Reassembled in: 166]
    Unreassembled Short Message fragment 1 of 4

070  00 00 00 9a 05 00 03 02  04 01 48 69 20 55 73 65   ··········Hi Use
080  72 2c 20 54 68 65 72 65  20 69 73 20 61 20 64 65   r, There  is a de
090  6d 61 6e 64 20 66 6f 72  20 53 4d 53 20 63 6f 6e   mand for  SMS con
0a0  6e 65 63 74 69 76 69 74  79 20 66 72 6f 6d 20 61   nectivit y from a
```

Fig 7.6.1 English lang with DCS zero

Here, we can see that **UDHI indicator is coming as 1**, which denotes it is a **multi-part** message. And in the **GSM SMS User Data**, we can see the fragments and it's details.

7.6.2 Data Coding in Other Language Case

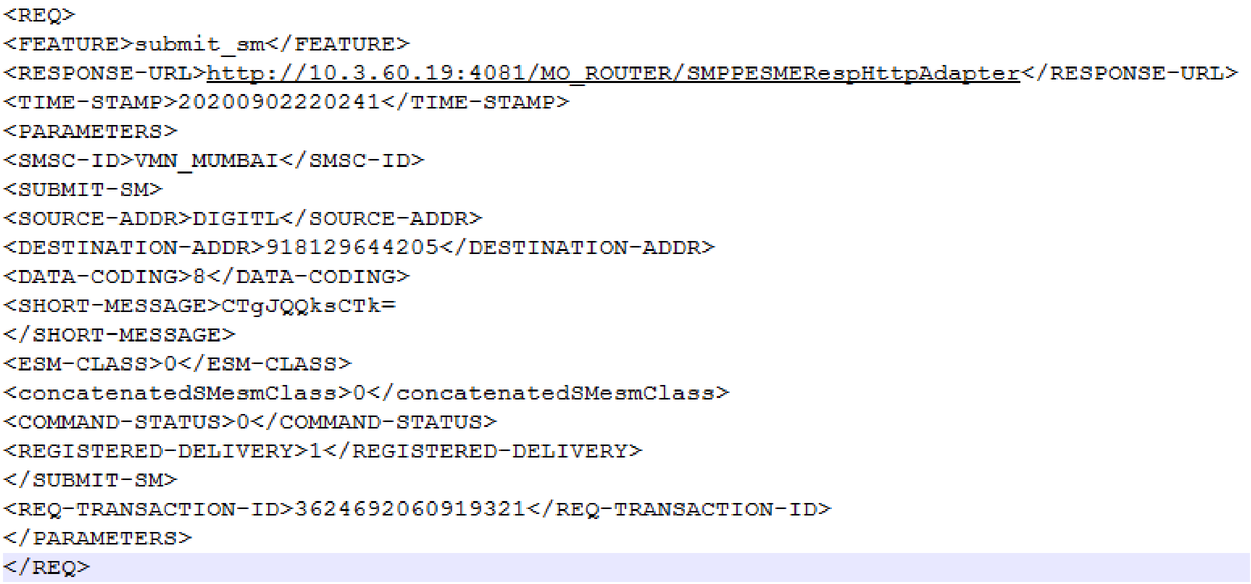
The ESME supports the other language script. The other language feature is often used in two different scenarios.

- Foreign language in English script(eg: *Indonesian in English script*) - Nomor yang Anda coba hubungi+6285771933138 saat ini sudah dalam jangkauan.
- Foreign language in own script(eg: *Arabic in Arabic script*) - ، ال م سـ تخدم مرحباًَ
الـ عربـ ية الأخرى لـ لغات DCS قـ يمة لاخـ تـ بار هذا.

First scenario can be achieved easily by replicating the same configuration we use for the English language, with **DCS zero**. But when we need the second scenario, the MO Router need to send the **DCS as 8 → UCS2**(most cases except French in some cases) in the request xml.

The MOR need to place the message string as **Decimal NCR equivalent** of required language script in their **message template** table. http://www.endmemo.com/unicode/unicodeconverter.php is one of the online converters we can use for this conversion.

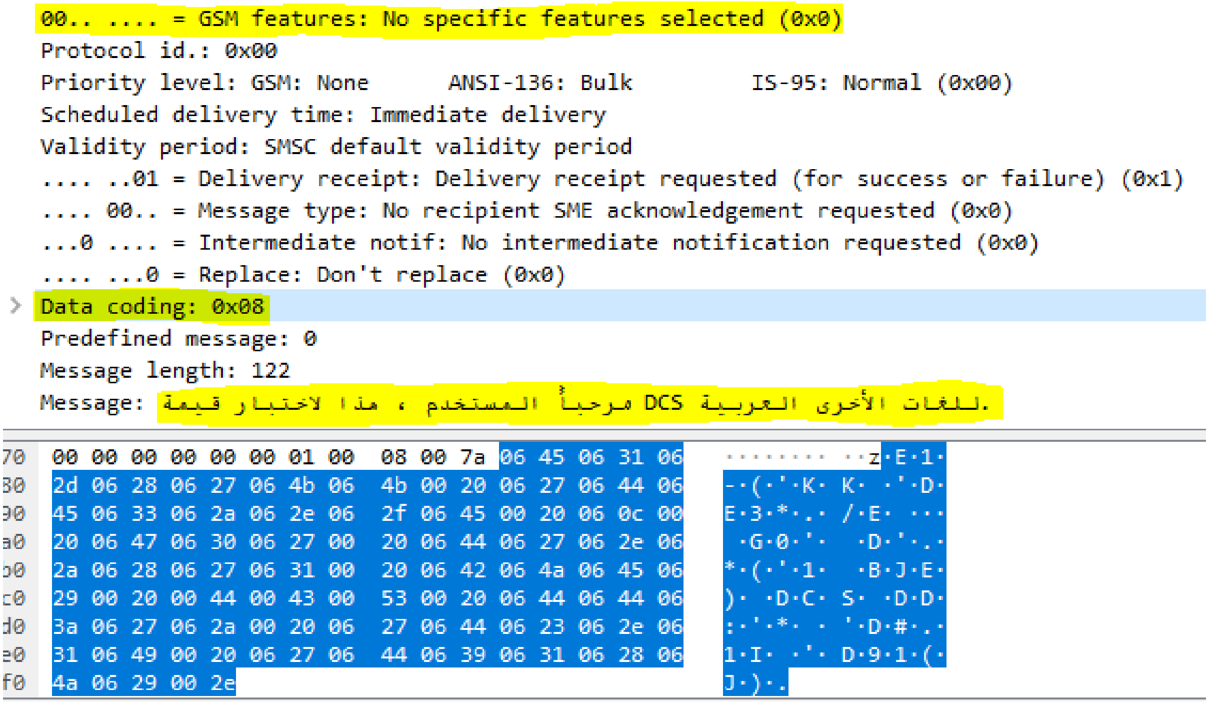
Now **MOR** needs to handle appropriate business logic to process this Decimal NCR string into base-64 encoded string [check **language converter class** for Arabic]. This base-64 encoded short message is passed to ESME via **SHORT-MESSAGE** tag in the request xml/json and ESME changes that into corresponding smpp equivalent (hex string) and sends to SMSC.



```
<REQ>
<FEATURE>submit_sm</FEATURE>
<RESPONSE-URL>http://10.3.60.19:4081/MO_ROUTER/SMPPESMERespHttpAdapter</RESPONSE-URL>
<TIME-STAMP>20200902220241</TIME-STAMP>
<PARAMETERS>
<SMSC-ID>VMN_MUMBAI</SMSC-ID>
<SUBMIT-SM>
<SOURCE-ADDR>DIGITL</SOURCE-ADDR>
<DESTINATION-ADDR>918129644205</DESTINATION-ADDR>
<DATA-CODING>8</DATA-CODING>
<SHORT-MESSAGE>CTgJQQksCTk=
</SHORT-MESSAGE>
<ESM-CLASS>0</ESM-CLASS>
<concatenatedSMesmClass>0</concatenatedSMesmClass>
<COMMAND-STATUS>0</COMMAND-STATUS>
<REGISTERED-DELIVERY>1</REGISTERED-DELIVERY>
</SUBMIT-SM>
<REQ-TRANSACTION-ID>3624692060919321</REQ-TRANSACTION-ID>
</PARAMETERS>
</REQ>
```

Fig 31: Sample request xml containg data-coding as 8

The most commonly used other langauge is Arabic. The trace for **Arabic langauge – small message** is available here.

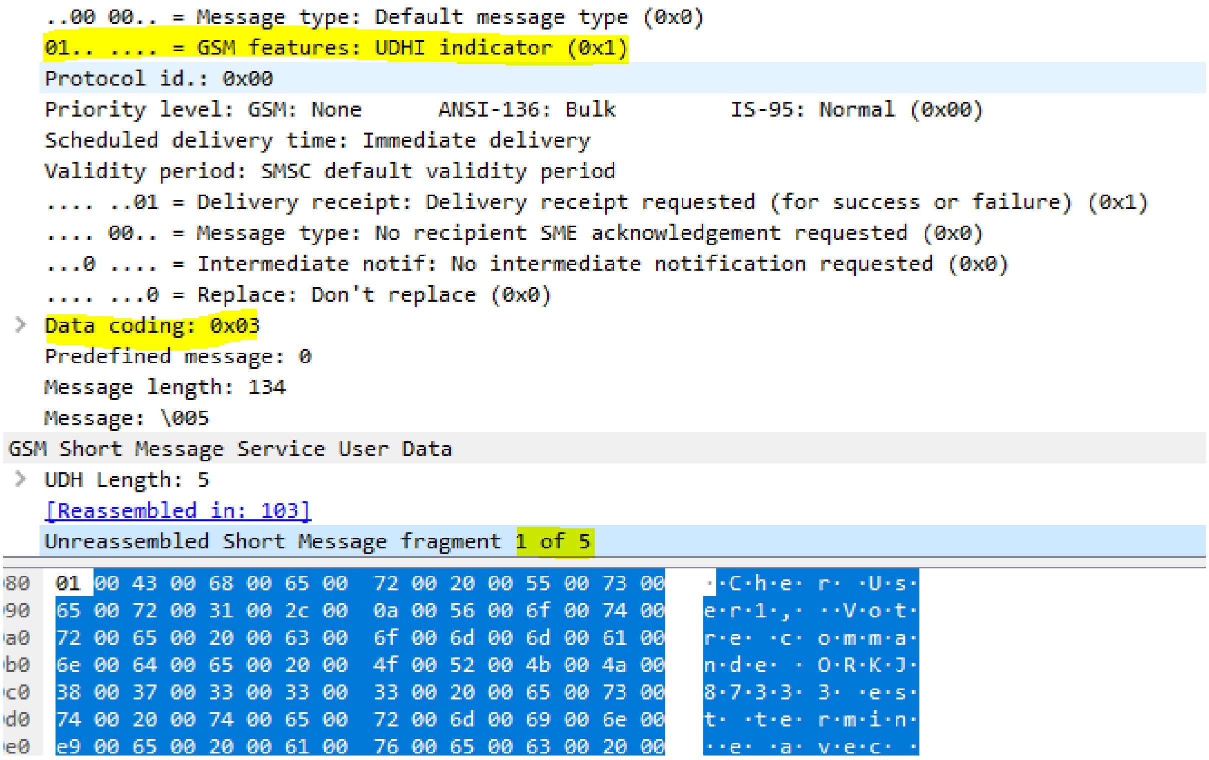


```
00.. .... = GSM features: No specific features selected (0x0)
Protocol id.: 0x00
Priority level: GSM: None      ANSI-136: Bulk        IS-95: Normal (0x00)
Scheduled delivery time: Immediate delivery
Validity period: SMSC default validity period
.... ..01 = Delivery receipt: Delivery receipt requested (for success or failure) (0x1)
.... 00.. = Message type: No recipient SME acknowledgement requested (0x0)
...0 .... = Intermediate notif: No intermediate notification requested (0x0)
.... ...0 = Replace: Don't replace (0x0)
Data coding: 0x08
Predefined message: 0
Message length: 122
Message: مرحباً المستخدم ، هذا لاختبار قيمة DCS للغات الأخرى العربية.

70  00 00 00 00 00 00 01 00  08 00 7a 06 45 06 31 06   ·········z·E·1·
30  2d 06 28 06 27 06 4b 06  4b 00 20 06 27 06 44 06   -·(·'·K· K· ·'·D·
90  45 06 33 06 2a 06 2e 06  2f 06 45 00 20 06 0c 00   E·3·*·.· /·E· ···
a0  20 06 47 06 30 06 27 00  20 06 44 06 27 06 2e 06    ·G·0·'·  ·D·'·.·
b0  2a 06 28 06 27 06 31 00  20 06 42 06 4a 06 45 06   *·(·'·1·  ·B·J·E·
c0  29 00 20 00 44 00 43 00  53 00 20 06 44 06 44 06   )· ·D·C· S· ·D·D·
d0  3a 06 27 06 2a 00 20 06  27 06 44 06 23 06 2e 06   :·'·*· · '·D·#·.·
e0  31 06 49 00 20 06 27 06  44 06 39 06 31 06 28 06   1·I· ·'· D·9·1·(·
f0  4a 06 29 00 2e                                      J·)·.
```

Fig 7.6.2 Arabic lang with DCS eight

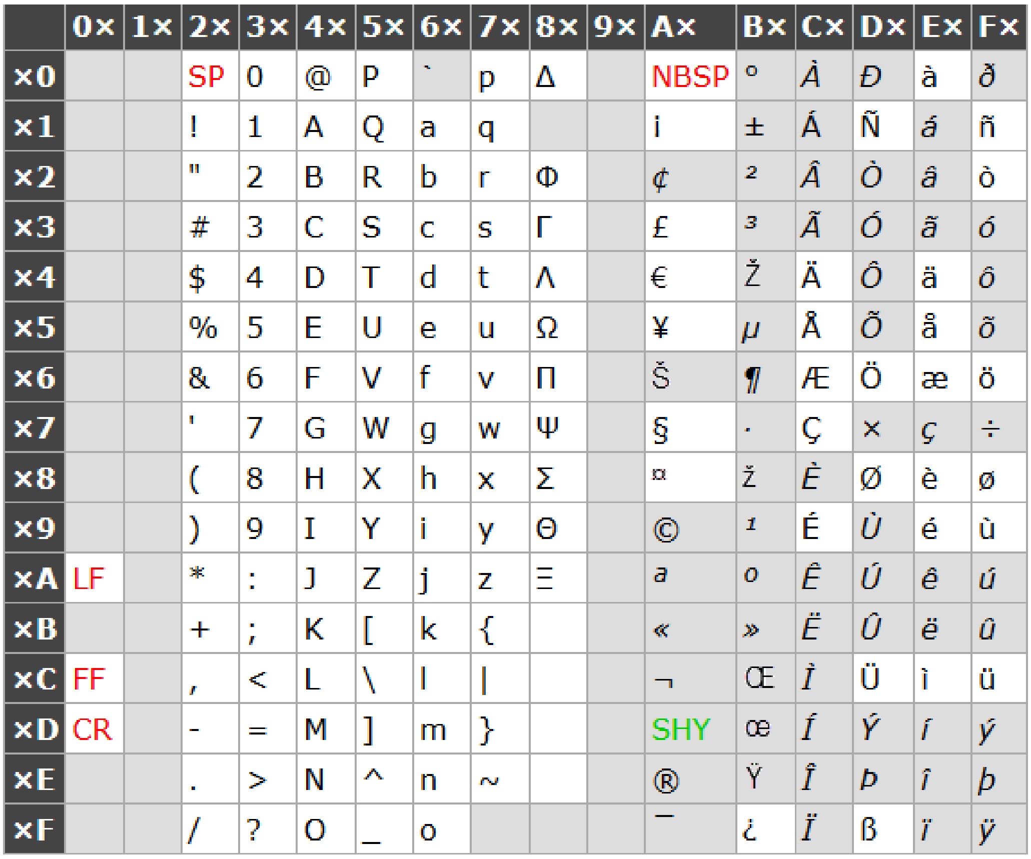
Since we are using a small message (less than 160 characters), there is no need for UDHI flag.

As we mentioned earlier, French lanaguage often uses **DCS as 3** (Latin 1/ ISO-8859-1 encoding*). The trace is available here.

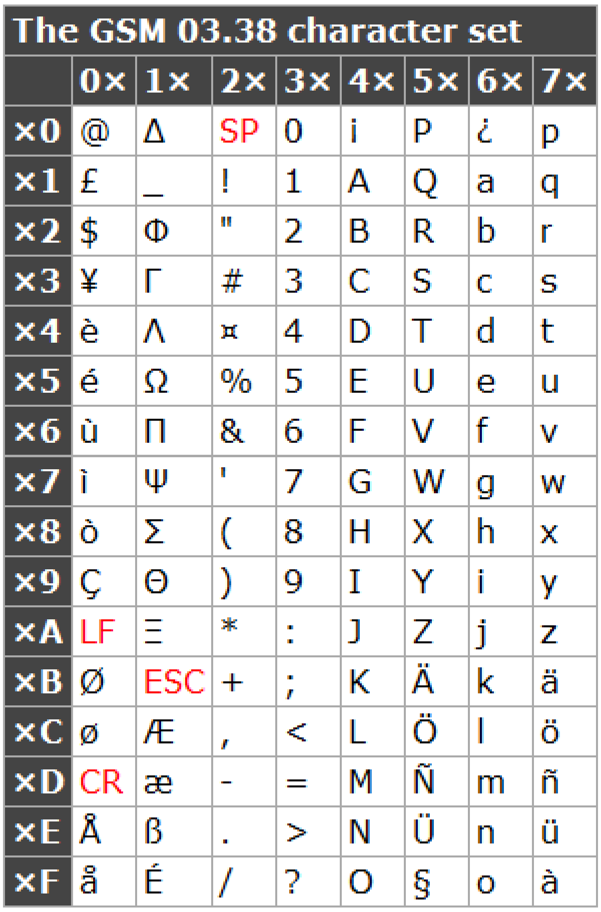


```
..00 00.. = Message type: Default message type (0x0)
01.. .... = GSM features: UDHI indicator (0x1)
Protocol id.: 0x00
Priority level: GSM: None      ANSI-136: Bulk        IS-95: Normal (0x00)
Scheduled delivery time: Immediate delivery
Validity period: SMSC default validity period
.... ..01 = Delivery receipt: Delivery receipt requested (for success or failure) (0x1)
.... 00.. = Message type: No recipient SME acknowledgement requested (0x0)
...0 .... = Intermediate notif: No intermediate notification requested (0x0)
.... ...0 = Replace: Don't replace (0x0)
Data coding: 0x03
Predefined message: 0
Message length: 134
Message: \005
GSM Short Message Service User Data
UDH Length: 5
[Reassembled in: 103]
Unreassembled Short Message fragment 1 of 5

80  01 00 43 00 68 00 65 00  72 00 20 00 55 00 73 00   ··C·h·e· r· ·U·s·
90  65 00 72 00 31 00 2c 00  0a 00 56 00 6f 00 74 00   e·r·1·,· ··V·o·t·
a0  72 00 65 00 20 00 63 00  6f 00 6d 00 6d 00 61 00   r·e· ·c· o·m·m·a·
b0  6e 00 64 00 65 00 20 00  4f 00 52 00 4b 00 4a 00   n·d·e· · O·R·K·J·
c0  38 00 37 00 33 00 33 00  33 00 20 00 65 00 73 00   8·7·3·3· 3· ·e·s·
d0  74 00 20 00 74 00 65 00  72 00 6d 00 69 00 6e 00   t· ·t·e· r·m·i·n·
e0  e9 00 65 00 20 00 61 00  76 00 65 00 63 00 20 00   ··e· ·a· v·e·c· ·
```

But the problem with above approach is that some of the French characters are not available in GSM 7 bit encoding. The characters not present in the GSM character set are shown on a **grey background**.



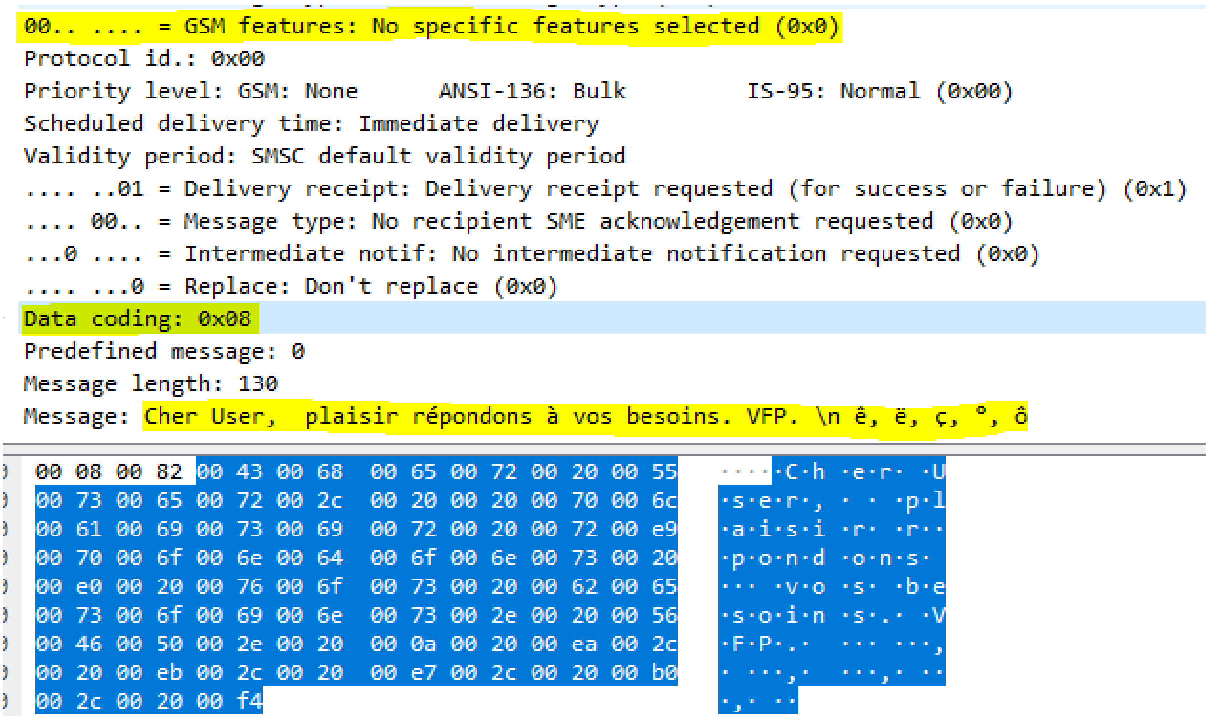
	0x	1x	2x	3x	4x	5x	6x	7x	8x	9x	Ax	Bx	Cx	Dx	Ex	Fx
x0			SP	0	@	P	`	p	Δ		NBSP	°	À	Ð	à	ð
x1			!	1	A	Q	a	q			¡	±	Á	Ñ	á	ñ
x2			"	2	B	R	b	r	Φ		¢	²	Â	Ò	â	ò
x3			#	3	C	S	c	s	Γ		£	³	Ã	Ó	ã	ó
x4			$	4	D	T	d	t	Λ		€	Ž	Ä	Ô	ä	ô
x5			%	5	E	U	e	u	Ω		¥	µ	Å	Õ	å	õ
x6			&	6	F	V	f	v	Π		Š	¶	Æ	Ö	æ	ö
x7			'	7	G	W	g	w	Ψ		§	·	Ç	×	ç	÷
x8			(	8	H	X	h	x	Σ		¤	ž	È	Ø	è	ø
x9			)	9	I	Y	i	y	Θ		©	¹	É	Ù	é	ù
xA	LF		*	:	J	Z	j	z	Ξ		ª	º	Ê	Ú	ê	ú
xB			+	;	K	[	k	{			«	»	Ë	Û	ë	û
xC	FF		,	<	L	\	l	|			¬	Œ	Ì	Ü	ì	ü
xD	CR		-	=	M	]	m	}			SHY	œ	Í	Ý	í	ý
xE			.	>	N	^	n	~			®	Ÿ	Î	Þ	î	þ
xF			/	?	O	_	o				¯	¿	Ï	ß	ï	ÿ



The GSM 03.38 character set

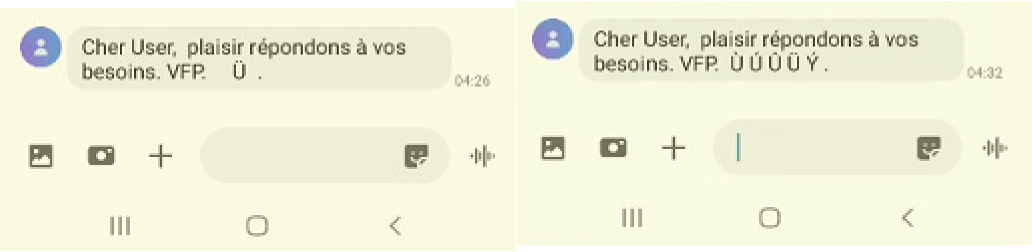
	0x	1x	2x	3x	4x	5x	6x	7x
x0	@	Δ	SP	0	¡	P	¿	p
x1	£	_	!	1	A	Q	a	q
x2	$	Φ	"	2	B	R	b	r
x3	¥	Γ	#	3	C	S	c	s
x4	è	Λ	¤	4	D	T	d	t
x5	é	Ω	%	5	E	U	e	u
x6	ù	Π	&	6	F	V	f	v
x7	ì	Ψ	'	7	G	W	g	w
x8	ò	Σ	(	8	H	X	h	x
x9	Ç	Θ	)	9	I	Y	i	y
xA	LF	Ξ	*	:	J	Z	j	z
xB	Ø	ESC	+	;	K	Ä	k	ä
xC	ø	Æ	,	<	L	Ö	l	ö
xD	CR	æ	-	=	M	Ñ	m	ñ
xE	Å	ß	.	>	N	Ü	n	ü
xF	å	É	/	?	O	§	o	à

If we want those grey characters, we need to use **DCS as 8(UC2 encoding)** just like Arabic.



```
00.. .... = GSM features: No specific features selected (0x0)
Protocol id.: 0x00
Priority level: GSM: None      ANSI-136: Bulk        IS-95: Normal (0x00)
Scheduled delivery time: Immediate delivery
Validity period: SMSC default validity period
.... ..01 = Delivery receipt: Delivery receipt requested (for success or failure) (0x1)
.... 00.. = Message type: No recipient SME acknowledgement requested (0x0)
...0 .... = Intermediate notif: No intermediate notification requested (0x0)
.... ...0 = Replace: Don't replace (0x0)
Data coding: 0x08
Predefined message: 0
Message length: 130
Message: Cher User,  plaisir répondons à vos besoins. VFP. \n ê, ë, ç, °, ô

 00 08 00 82 00 43 00 68  00 65 00 72 00 20 00 55   ·····C·h ·e·r· ·U
 00 73 00 65 00 72 00 2c  00 20 00 20 00 70 00 6c   ·s·e·r·, · · ·p·l
 00 61 00 69 00 73 00 69  00 72 00 20 00 72 00 e9   ·a·i·s·i ·r· ·r··
 00 70 00 6f 00 6e 00 64  00 6f 00 6e 00 73 00 20   ·p·o·n·d ·o·n·s·
 00 e0 00 20 00 76 00 6f  00 73 00 20 00 62 00 65   ··· ·v·o ·s· ·b·e
 00 73 00 6f 00 69 00 6e  00 73 00 2e 00 20 00 56   ·s·o·i·n ·s·.· ·V
 00 46 00 50 00 2e 00 20  00 0a 00 20 00 ea 00 2c   ·F·P·.·  ··· ···,
 00 20 00 eb 00 2c 00 20  00 e7 00 2c 00 20 00 b0   · ···,·  ···,· ··
 00 2c 00 20 00 f4                                  ·,· ··
```

Fig 7.6.2 French lang with DCS eight



Cher User, plaisir répondons à vos besoins. VFP.   Ü .

Cher User, plaisir répondons à vos besoins. VFP. Ù Ú Û Ü Ý .

Fig 7.6.2 Handset Snapshot French lang with DCS three & eight

Note:

- Foreign language in their own script can't be place directly in the message templates table at the MOR side. Only English script works like the above.
- ESME don't have any business logic regarding conversion of decimal NCR into equivalent base 64 encoded short message.
- For DCS = 3, MOR originally used utf-16 encoding, which will take two bytes for each character. Later they changed into ISO-8859-1 encoding, which is a single byte encoding, that helps to send more GSM7 characters.

8 Error Codes

8.1 ESME Response Codes

ErrorCode	ErrorDesc
-1001	Unknown feature or Illegal feature
-1002	Overload rejection
-1003	Internal server error (check feature_id)
-1004	Unknown Operation or Exception
1000	Illegal request Message
1005	Mandatory field missing
1006	Illegal data type
1007	Username or password is not match
1008	Other Error i.e. Not able to send response
1009	Unknown CommandId
1010	Execute Exception
1011	CMD_FIELD_MISSING_ERRORCODE
1012	UNKNOWN_COMMANDID_ERRORCODE
1013	UNKNOWN_SMSC_ERRORCODE
1014	NOT_BOUND_WITH_SMSC_ERRORCODE
1015	ALREADY_BOUND_WITH_SMSC_ERRORCODE
1016	IOEXCEPTION_ERRORCODE
1017	NEGATIVE_RESPONSE_ERRORCODE
1018	RESPONSE_TIMEOUT_ERRORCODE
1019	UNSUC_DELIVERY_SME_ERRORCODE
1020	THROTTLING_LIMIT_ERRORCODE

Figure 32: ESME Response Codes

8.2 SMSC Error Codes

VALUE	DESCRIPTION
0x00000000	No Error
0x00000001	Message Length is invalid
0x00000002	Command Length is invalid
0x00000003	Invalid Command ID
0x00000004	Incorrect BIND Status for given command
0x00000005	ESME Already in Bound State
0x00000006	Invalid Priority Flag
0x00000007	Invalid Registered Delivery Flag
0x00000008	System Error
0x00000009	Reserved
0x0000000A	Invalid Source Address
0x0000000B	Invalid Destination Address
0x0000000C	Message ID is invalid
0x0000000D	Bind Failed
0x0000000E	Invalid Password
0x0000000F	Invalid System ID
0x00000010	Reserved
0x00000011	Cancel SM Failed
0x00000012	Reserved
0x00000013	Replace SM Failed
0x00000014	Message Queue Full
0x00000015	Invalid Service Type
0x00000016	
0x00000032	Reserved
0x00000033	Invalid number of destinations
0x00000034	Invalid Distribution List name

VALUE	DESCRIPTION
0x00000035	
0x0000003F	Reserved
0x00000040	Destination flag is invalid (submit_multi)
0x00000041	Reserved
0x00000042	Invalid 'submit with replace' request (e.g. submit_sm with replace_if_present_flag set)
0x00000043	Invalid esm_class field data
0x00000044	Cannot Submit to Distribution List
0x00000045	submit_sm or submit_multi failed
0x00000046	
0x00000047	Reserved
0x00000048	Invalid Source address TON Check the document for correct TON/NPI values
0x00000049	Invalid Source address NPI
0x00000050	Invalid Destination address TON
0x00000051	Invalid Destination address NPI
0x00000052	Reserved
0x00000053	Invalid system_type field
0x00000054	Invalid replace_if_present flag
0x00000055	Invalid number of messages
0x00000056	
0x00000057	Reserved
0x00000058	Throttling error (ESME has exceeded allowed message limits)
0x00000059	
0x00000060	Reserved
0x00000061	Invalid Scheduled Delivery Time
0x00000062	Invalid message validity period (Expiry time)
0x00000063	Predefined Message Invalid or Not Found

VALUE	DESCRIPTION
0x00000064	ESME Receiver Temporary App Error Code
0x00000065	ESME Receiver Permanent App Error Code
0x00000066	ESME Receiver Reject Message Error Code
0x00000067	query_sm request failed
0x00000068	
0x000000A9	
0x000000BF	Reserved
0x000000C0	Error in the optional part of the PDU Body
0x000000C1	Optional Parameter not allowed
0x000000C2	Invalid Parameter Length.
0x000000C3	Expected Optional Parameter missing
0x000000C4	Invalid Optional Parameter Value
0x000000C5	
0x000000FD	Reserved
0x000000FE	Delivery Failure (used for data_sm_resp)
0x000000FF	Unknown Error
0x00000100	
0x000003FF	Reserved for SMPP extension
0x00000400	
0x000004FF	Reserved for SMSC vendor specific errors
0x00000500	
0xFFFFFFFF	ReservedMessage State

Figure 33: SMSC Error Codes