General ESME Questions

What is ESME?
ESME stands for External Short Messaging Entity, a module used for sending and receiving SMS messages.

What is the purpose of the ESME module?
The ESME module is used for interfacing with an SMS gateway to facilitate SMS communication.

Why is ESME important?
ESME is important for applications that require SMS functionality, such as notifications, alerts, and user authentication.

WildFly Setup Questions

What version of WildFly is required for ESME?
A minimum of WildFly 16 is required.

What JDK version is required for WildFly when setting up ESME?
JDK 1.8 is required.

How do you configure JDK in WildFly?
You set the JAVA_HOME environment variable to the JDK installation path:

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

Where should JAVA_HOME be set for WildFly?
It should be set in the environment variables.

What happens if the required JDK version is not installed?
WildFly may not function properly, causing compatibility issues with ESME.

Memory Configuration in WildFly

Where can you configure memory settings in WildFly?

In the standalone.conf file located in the bin directory of the WildFly installation.

How do you set memory allocation for WildFly?

Modify the -Xmx option in the standalone.conf file:

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

This sets the heap memory allocation.

Why is memory configuration important in WildFly?

Proper memory allocation ensures optimal performance and prevents OutOfMemoryErrors.

What does -Xms and -Xmx mean in WildFly configuration?
-Xms sets the initial heap size, and -Xmx sets the maximum heap size.

Standalone.xml Configuration

Where is the WildFly configuration file located?

The configuration file is located at:

```
$WILDFLY_HOME/standalone/configuration/standalone.xml
```

Which section of standalone.xml contains port configurations?

The <socket-binding-group> section.

What are the key ports configured in standalone.xml?

AJP Port
HTTPS Port
HTTP Port (default is 8080)

How can you change the HTTP port in WildFly?

Modify the <socket-binding> configuration inside standalone.xml and set a different port.

What is the default HTTP port in WildFly?
The default HTTP port is 8080.

How can you check if a port is available before configuring it?

Use the netstat command:

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

Why should you check port availability before modifying WildFly ports?

To avoid conflicts with other running applications.

How do you configure the management HTTP and HTTPS ports in WildFly?

The management ports are defined in the <socket-binding-group> of standalone.xml.

Command-Based Questions

What command checks if port 8080 is already in use?

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

How do you find the WildFly home directory?

Typically, it is set in an environment variable like $WILDFLY_HOME.

How can you restart WildFly after changing the configuration?

Run the following command:

```
./standalone.sh --server-config=standalone.xml
```

How do you check if WildFly is running?

Use:

```
ps aux | grep wildfly
```

Possible Troubleshooting Q&A

What should you do if WildFly does not start after modifying memory settings?

Check the standalone.conf file for syntax errors and verify available system memory.
What could cause a port conflict when starting WildFly?

Another application might be using the same port. Use netstat to check.
What should you do if HTTP port 8080 is unavailable?

Change it in standalone.xml to an available port.
Why might ESME fail to communicate with WildFly?

Possible reasons include incorrect port configuration, insufficient memory, or an unsupported JDK version.

What is Esme.war?

Esme.war is a deployment package that includes configuration files, essential libraries, and servlet mappings for WildFly.

What are the components of Esme.war?

The Esme.war package includes:
classes – Contains configuration files like log4j.xml and smpp.xml.
lib – Holds essential library files.
web.xml – Defines and maps servlets.

How can Esme.war be deployed?

Esme.war can be deployed using:
Admin Panel – Upload the file via the WildFly Admin Console in the Deployment section.
'dodeploy' Marker – Place Esme.war in $WILDFLY_HOME/standalone/deployments/ and create Esme.war.dodeploy to trigger automatic deployment.

Deployment Process Questions

What is the WildFly Admin Panel method of deploying Esme.war?

Access the WildFly Admin Console, go to the Deployment section, and use the provided options to upload and deploy Esme.war.
How does the 'dodeploy' marker method work for deploying Esme.war?

Place Esme.war inside $WILDFLY_HOME/standalone/deployments/ and create a Esme.war.dodeploy file in the same directory. WildFly will detect the marker file and automatically deploy the application.

What is the purpose of the Esme.war.dodeploy file?

It acts as a trigger for WildFly to recognize and deploy Esme.war when placed in the deployment directory.

Where should Esme.war be placed for deployment using the 'dodeploy' method?

The file should be placed in $WILDFLY_HOME/standalone/deployments/.

What happens if the Esme.war.dodeploy file is missing?

WildFly will not automatically deploy the Esme.war file.

Database Driver Questions

How do you add MySQL or Oracle JDBC driver to WildFly?

The driver must be added to the respective class path in WildFly.
Where should the MySQL JDBC driver be placed in WildFly?

The MySQL driver should be placed in:

$WILDFLY_HOME/modules/system/layers/base/com/mysql/jdbc/Driver/main

What additional file needs to be created when adding the MySQL JDBC driver?
A module.xml file must be created in the same directory.

What should the module name in module.xml match?
The module name should match the path where the driver is placed.

What should the resource-root-path in module.xml contain?
It must specify the name of the MySQL JAR file.

Can Esme.war be deployed without accessing the Admin Console?
Yes, by using the ‘dodeploy’ marker method.

Why do we need to create a module.xml file when adding the MySQL driver?
WildFly requires module.xml to recognize and load the MySQL driver as a module.

What happens if the module name in module.xml does not match the path?
WildFly will fail to locate and load the MySQL driver.

What will happen if Esme.war is placed in the deployments directory without a marker file?
WildFly will not deploy it unless the deployment is triggered manually.

Is it necessary to restart WildFly after adding the MySQL driver?
Yes, a restart ensures that the new driver module is loaded correctly.

Configuring JNDI for MySQL/Oracle in WildFly

General Questions
What is the purpose of configuring JNDI for MySQL or Oracle in WildFly?

Configuring JNDI allows applications deployed on WildFly to access MySQL or Oracle databases using JDBC drivers.

What are the main steps to integrate MySQL or Oracle with WildFly?

The integration involves placing the JDBC driver in the appropriate directory, creating a module.xml file, and restarting WildFly.

MySQL Integration

Where should the MySQL JDBC JAR file be placed in WildFly?

It should be placed in:

$WILDFLY_HOME/modules/system/layers/base/com/mysql/jdbc/Driver/main
What is the purpose of the module.xml file in MySQL integration?

It defines the module for the MySQL JDBC driver, specifying the driver location and dependencies.

What dependencies are required for MySQL module configuration?

The dependencies include:

```
<dependencies>
   <module name="javax.api"/>
   <module name="javax.transaction.api"/>
</dependencies>
```

How should the module.xml file be structured for MySQL?

The structure is:

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.3" name="com.mysql.jdbc.Driver">
   <resources>
      <resource-root path="mysql-connector-java-{version}.jar"/>
   </resources>
   <dependencies>
      <module name="javax.api"/>
      <module name="javax.transaction.api"/>
   </dependencies>
</module>
```

{version} should be replaced with the actual MySQL JDBC driver version.

Oracle Integration

Where should the Oracle JDBC JAR file be placed in WildFly?

It should be placed in a similar structure as MySQL but in an Oracle-specific directory.
What changes need to be made to module.xml for Oracle JDBC integration?

The module.xml should reference the Oracle JDBC JAR file:

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.3" name="com.oracle.jdbc">
  <resources>
    <resource-root path="ojdbc{version}.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

{version} should be replaced with the actual Oracle JDBC driver version.

What is the final step after adding the JDBC driver and module.xml?

Restart WildFly to apply the changes.

Logging in WildFly

How is logging managed in WildFly?

Logging in WildFly is handled through the logging subsystem, which includes handlers, loggers, root logger declarations, and logging profiles.

What are the four major components of WildFly's logging subsystem?

The components are:
Handler configurations
Loggers
Root logger declarations
Logging profiles

Per-Deployment Logging

What is per-deployment logging in WildFly?

It allows each deployment to have a separate logging configuration by including a logging configuration file within the deployment package.

Where should a logging configuration file be placed in a WAR deployment?

It should be placed inside the WEB-INF/classes directory.

Supported Logging Configuration Files

What are the acceptable logging configuration file formats in WildFly?
The supported formats are:
logging.properties
jboss-logging.properties
log4j.properties
log4j.xml
jboss-log4j.xml

Q: What is the purpose of the smpp.xml file in ESME?
A: The smpp.xml file is used to configure ESME settings, ensuring a successful connection and communication with the SMSC (Short Message Service Centre).

Q: Where is the smpp.xml file located?
A: The smpp.xml file is located at $WILDFLY_HOME/standalone/deployments/ESME.war/WEB-INF/classes/smpp.xml.

Q: What must be verified in the smpp.xml file to ensure proper ESME functionality?
A: The file must be checked for accurate configurations of parameters like SMSC ID, IP address, port, system ID, password, bind mode, and registered delivery settings.

SMSC Configuration Parameters
Q: What is the purpose of the smsc id parameter?
A: It provides a unique identifier for the SMSC.

Q: Why is the IP address parameter important in smpp.xml?
A: It defines the IP address of the SMSC, ensuring the ESME connects to the correct server.

Q: What does the port parameter specify?
A: It specifies the SMPP port used for communication with the SMSC.

Q: What is the function of the system-id parameter?
A: The system-id identifies the ESME system requesting to bind as a transceiver with the SMSC.

Q: What is the password requirement according to the SMPP standard?
A: The password should be less than 9 characters.

Binding and Message Delivery Settings
Q: What is the bind mode used for in smpp.xml?
A: The bind mode defines the type of connection ESME establishes with SMSC:

‘t’ for transmitter
‘r’ for receiver
‘tr’ for transceiver
Q: What is the registered delivery parameter?
A: It determines whether the SMSC should send a delivery receipt for messages.

Q: What are the possible values for registered_delivery?
A:

1 → Delivery receipt is required.
0 → No delivery receipt is required.
SMPP Protocol Version and Interface Settings
Q: What should be the interface-version in smpp.xml?
A: It should be equal to the ASCII value of the SMPP version. The current SMPP version is 3.4, which corresponds to ASCII value 4.

TON (Type of Number) and **NPI** (Numbering Plan Indicator) Configuration
Q: What are the TON and NPI values used for?
A: They define the type and numbering plan used for source and destination addresses in SMS messaging.

Q: What are the possible values for TON?
A:

0 → Unknown
1 → International

2 → National
3 → Network Specific
4 → Subscriber Number
5 → Alphanumeric (e.g., a word with numbers)
6 → Abbreviated
Q: What are the possible values for NPI?
A:

0 → Unknown
1 → ISDN/Telephone numbering plan (E.163/E.164)
3 → Data numbering plan (X.121)
4 → Telex numbering plan (F.69)
6 → Land Mobile (E.212)
8 → National numbering plan
9 → Private numbering plan
10 → ERMES numbering plan (ETSI DE/PS 3 01-3)
13 → Internet (IP)
18 → WAP Client ID (Defined by WAP Forum)
Q: What does a TON value of 5 indicate?
A: It indicates an alphanumeric address, meaning the sender address contains letters and numbers.

Q: What does an NPI value of 1 indicate?
A: It indicates the ISDN/telephone numbering plan (E.163/E.164).

What is the purpose of the Connector section?
A: The Connector section details the transport protocol, transport method, transport URLs, and message format used to connect to the MO_Router or any other application in case of an incoming feed from SMSC.

Q: What file contains details about the Connector section?
A: The details about the Connector section are available in the SMPP.xml file.

Q: What is the format of the Connector XML configuration?
A: The Connector XML configuration includes parameters like max_connection, transportprotocol, transportmethod, transporturl1, transporturl2, and messageformat.

Q: What does the <max_connection> field in the Connector XML define?
A: The <max_connection> field defines the number of threads that send messages to NG/third-party services through HTTP. The value can range from 1 to 5.

Q: What are the possible values for <transportprotocol> in the Connector XML?
A: The <transportprotocol> field specifies the transport protocol, such as http.

Q: What is the purpose of the <transportmethod> field in the Connector XML?
A: The <transportmethod> field specifies the transport method used for communication, such as post.

Q: What do <transporturl1> and <transporturl2> define in the Connector XML?
A: <transporturl1> and <transporturl2> specify the URLs where the request XML should be sent.

Q: What is the purpose of <messageformat> in the Connector XML?
A: The <messageformat> defines the format in which the request XML is sent from the ESME to the MO_Router (MOR).

Q: How does <MESSAGE> differ from <SMS> in multi-part messages?
A: <MESSAGE> contains the decoded message string from <SMS-BYTE>, after removing UDH headers. <SMS> may contain unreadable UDH headers at the beginning of the message.

Q: What is the role of <MESSAGE-IDENTIFIER> in multi-part messages?
A: <MESSAGE-IDENTIFIER> provides a unique identifier that helps concatenate multi-part messages at the MO_Router side.

Q: What does <MESSAGE-PARTS> indicate in an XML message?
A: <MESSAGE-PARTS> represents the total number of parts in a multi-part message.

Q: What does <MESSAGE-PART-NUMBER> indicate?
A: <MESSAGE-PART-NUMBER> provides the sequence number of the message part, helping to reconstruct the original message in the correct order.

Q: What happens when a message length is less than 160 characters?

A: When a message is less than 160 characters, the ESM-CLASS value is set to 0, and no multi-part messages are generated. The <MESSAGE-IDENTIFIER>, <MESSAGE-PARTS>, and <MESSAGE-PART-NUMBER> fields remain empty.

Q: What is the significance of ESM-CLASS in message processing?
A: ESM-CLASS determines whether a message is a single-part or multi-part message. A value of 0 indicates a single-part message, while a value of 64 indicates a multi-part message.

Q: How does the system handle multi-part messages from SMSC?
A: Multi-part messages from SMSC are split into separate packets. <MESSAGE-IDENTIFIER> remains the same across all parts, while <MESSAGE-PART-NUMBER> determines the sequence, and <MESSAGE-PARTS> provides the total count.

Q: What happens when ESM-CLASS is set to 64?
A: When ESM-CLASS is 64, the message is a multi-part message, meaning it will be split into multiple parts with identifiers to concatenate them later.

Q: What does <SMS-BYTE> contain?
A: <SMS-BYTE> holds the encoded version of the message, which includes UDH headers in multi-part messages.

Q: What happens when UDH headers are present in a message?
A: When UDH headers are present, unwanted characters may appear in <SMS>. These are removed in <MESSAGE> to provide the clean decoded message.

Q: How does <TIME-STAMP> function in the XML structure?
A: <TIME-STAMP> records the time at which the message was processed.

Q: What is <TRANSACTION-ID> used for in the XML?
A: <TRANSACTION-ID> uniquely identifies a transaction in the message flow.

Q: What does <SMPP-ID> represent?
A: <SMPP-ID> indicates the source of the message, such as the SMSC or service provider name.

Q: What is the role of <OA> and <DA> in the XML message?
A: <OA> represents the originating address (sender), and <DA> represents the destination address (receiver) of the message.

Q: What does <DCS> define in the XML?
A: <DCS> (Data Coding Scheme) specifies the encoding format of the message.

Q: What happens when a message contains a newline character?
A: If a message contains a newline character, it is processed based on the transport protocol and method, and multi-part handling may apply depending on its length.

Q: What is the purpose of the MESSAGE-IDENTIFIER, MESSAGE-PARTS, and MESSAGE-PART-NUMBER fields?
A: These fields are empty because no UDH headers are coming from the SMSC.

Q: What does the delivery connector do?
A: The delivery connector details the transport protocol, transport method, transport URLs, and message format used to send the delivery receipt when RegisteredDelivery is set to 1, to a third-party module like UMS/NG.

Q: Where can I find more details about the delivery connector?
A: More details about the delivery connector can be found in the SMPP.xml documentation.

Delivery Connector XML Questions

Q: What is the purpose of the <max_connection> tag in the delivery connector XML?
A: It specifies the number of threads that send messages to NG/third-party through HTTP. The range can be between 1 and 5.

Q: What transport protocol is used in the delivery connector?
A: The transport protocol used is HTTP.

Q: What is the role of <transporturl1> and <transporturl2> in the delivery connector XML?
A: These URLs specify the endpoints to which the request XML should be sent.

Q: What does the <messageformat> tag define in the delivery connector XML?

A: It defines the format in which the request XML is sent from the ESME to the MOR when RegisteredDelivery is set to 1.

Q: What is the purpose of the MESSAGE-IDENTIFIER, MESSAGE-PARTS, and MESSAGE-PART-NUMBER fields?
A: These fields are empty because no UDH headers are coming from the SMSC.

Q: What does the delivery connector do?
A: The delivery connector details the transport protocol, transport method, transport URLs, and message format used to send the delivery receipt when RegisteredDelivery is set to 1, to a third-party module like UMS/NG.

Q: Where can I find more details about the delivery connector?
A: More details about the delivery connector can be found in the SMPP.xml documentation.

Delivery Connector XML Questions
Q: What is the purpose of the <max_connection> tag in the delivery connector XML?
A: It specifies the number of threads that send messages to NG/third-party through HTTP. The range can be between 1 and 5.

Q: What transport protocol is used in the delivery connector?
A: The transport protocol used is HTTP.

Q: What is the role of <transporturl1> and <transporturl2> in the delivery connector XML?
A: These URLs specify the endpoints to which the request XML should be sent.

Q: What does the <messageformat> tag define in the delivery connector XML?
A: It defines the format in which the request XML is sent from the ESME to the MOR when RegisteredDelivery is set to 1.

Sample Message Format Questions

Q: What is the structure of a sample delivery receipt message?
A: A sample delivery receipt message is structured as follows:

```
<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>
```

Q: What does the <FEATURE> tag represent in the message format?
A: It represents the feature type, which in this case is DELIVERY-RECEIPT.

Q: What is the significance of the <ESM-CLASS> tag in the sample message?
A: The value of ESM-CLASS is set to 4, which indicates that this is a delivery receipt message.

Q: What information is included in the <SMS> tag of the message format?
A: It contains details such as message ID, submission status, delivery status, error code, and text of the SMS.

Q: What is the meaning of stat:DELIVRD in the <SMS> tag?
A: It indicates that the SMS has been successfully delivered.

Q: What is the role of <MESSAGE-ID> in the message format?
A: It uniquely identifies the message.

Optional Parameters Questions

Q: What are optional parameters in the delivery receipt message?
A: Optional parameters are additional tags used for special cases like message payloads.

Q: What is an example of an optional parameter tag?
A:

```
<optionalparameterstag>
  <tlv1>
    <tag>5126</tag>
    <name>OptionalParam_OA</name>
  </tlv1>
</optionalparameterstag>
```

Here, 5126 is the tag identifier, and OptionalParam_OA is the name of the optional parameter

Service-type

Q: What is the purpose of the service_type parameter?
A: The service_type parameter indicates the SMS Application service associated with the message.

Q: What are the benefits of specifying the service_type parameter?
A: Specifying service_type allows the ESME to:

Avail of enhanced messaging services, such as replace by service type.
Control the tele-service used on the air interface.
Q: What should be the value of service_type for default SMSC settings?
A: The service_type should be set to NULL for default SMSC settings.

Dcs (Data Coding Scheme)

Q: What is Dcs in SMS messaging?
A: Dcs stands for Data Coding Scheme, which defines the encoding scheme of the short message user data.

Q: What is the default Dcs value for English messages?
A: The default Dcs value for English is 0.

Q: What Dcs value is used for other languages?
A: The Dcs value used for other languages is 8.

Q: What is the Dcs value for Flash Messages?
A: The Dcs value for Flash Messages is 16.

messageReceiverListenerImpl

Q: What does the messageReceiverListenerImpl tag represent?
A: The messageReceiverListenerImpl tag provides the full class path to the class messageReceiverListenerImpl.

Q: What is the current value of messageReceiverListenerImpl?
A: The current value is com.sixdee.imp.receiver.MessageReceiverListenerImpl.

Q: When should the value of messageReceiverListenerImpl be changed?
A: The value should only be changed if the Development Team suggests so.

TPS (Transactions Per Second)

Q: What is TPS in the context of SMS messaging?
A: TPS (Transactions Per Second) is the number of transactions executed per second.

Q: How is TPS calculated?
A: TPS is calculated by dividing the total number of transactions over a certain duration by the number of seconds.
For example, if a user executes 6 transactions per minute, TPS = 6 / 60 = 0.10 TPS.

Q: What is the default TPS value?
A: The default value is -1.

Q: When should TPS be set to 1?
A: The TPS should be set to 1 when ESME is implemented with retry.

enquireLinkTimer
Q: What is enquireLinkTimer?
A: enquireLinkTimer is an SMPP session timer that allows the ESME to request the SMPP session status via the enquire_link command.

Q: What is the range of enquireLinkTimer values?
A: The value ranges from 5 to 10 seconds.

Receive-timeout

Q: What is receive-timeout?
A: receive-timeout is the timeout for attempting to receive a message from the SMSC in a synchronized way.

Q: What is the valid range for receive-timeout?
A: The valid range is 5 to 10 seconds.

Pdu-processors

Q: What does pdu-processors represent?
A: pdu-processors defines the number of threads processing SMSC requests simultaneously.

Q: What is the maximum value of pdu-processors?
A: The maximum value is 5.

pdu-maxQueueSize

Q: What is pdu-maxQueueSize?
A: pdu-maxQueueSize sets the waiting queue size for pdu-processors.

Q: How to set pdu-maxQueueSize to infinite?
A: Set pdu-maxQueueSize to -1 to make it infinite.

Q: How to set pdu-maxQueueSize to a specific limit?
A: Set pdu-maxQueueSize to the required numerical limit.

allowedNumSeries

Q: What does allowedNumSeries specify?

A: allowedNumSeries lists the allowed number series for processing.

Q: What is the default value of allowedNumSeries?
A: The default value is "*" which means "all" numbers are allowed.

Q: What is the URL for the HttpAdapter?
A: http://localhost:8081/ESME/HttpAdapter

Q: What is the URL for the HttpNoRespAdapter?
A: http://localhost:8081/ESME/HttpNoRespAdapter

submit_sm sample packet is below :

```
 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>
```

Synch_Submit_sm sample packet is below :

```
 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>SSBjYWxsZWQgWW91IGF0IFdlZCwgOSBTZXAgMjAxNSAxODoxMDoyN
Q==</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>
```

MESSAGE PAYLOAD & UDH

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>SSBjYWxsZWQgWW91IGF0IFdlZCwgOSBTZXAgMjAxNSAxODoxMDoyN
Q==</SHORT-MESSAGE>
<DATA-CODING>0</DATA-CODING>
 </SUBMIT-SM>
 </PARAMETERS>
 </REQ>
```

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>SSBjYWxsZWQgWW91IGF0IFdlZCwgOSBTZXAgMjAxNSAxODoxMDoyN
Q==</SHORT-MESSAGE>
```

```
<DATA-CODING>0</DATA-CODING>
<MESSAGE-PAYLOAD>true</MESSAGE-PAYLOAD>
 </SUBMIT-SM>
 </PARAMETERS>
 </REQ>
```

ESME MO & Delivery Packets [in smpp.xml]

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[IEhlbGxvIGZyb20gU01QUFNpbSBIZWxsbyBmcm9tIFNNUFBTaW0gSGVsbG8gZnJvbSBTTVBQU2ltIEhlbGxvIGZyb20gU01QUFNpbUhlbGxvIGZyb20gU01QUFNpbUhlbGxvIGZyb20gU01QUFNpbUhlbGxvIGZyb20gU01QUFNpbQ==]]></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>
```

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>
```

Q: What is the featureId in the XML request?
A: FEATURE (string) - Feature of the ESME.

Q: What is the timeStamp in the XML request?
A: TIME-STAMP (timestamp) - Time stamp of the request.

Q: What is reqTransactionId in the XML request?
A: REQ-TRANSACTION-ID (integer) - Transaction ID of the request.

Q: What is the respUrl in the XML request?
A: RESPONSE-URL (string) - URL to which the response of the submit_sm feature should be sent. Not needed in synch_submit_sm.

Q: What is submitSm in the XML request?
A: SUBMIT-SM - The submit short message feature, which holds the message data.

Q: What is smscId in the XML request?
A: SMSC-ID (string) - Unique ID used to identify the specific SMSC.

Q: What is sourceAddr in the XML request?
A: SOURCE-ADDR (Variable, max 21, C Octet String) - Address of SME which originated this message.

Q: What is sourceAddrTon in the XML request?
A: SOURCE-ADDR-TON (integer, 1) - Type of number for source address. If not known, set to NULL (Unknown).

Q: What is sourceAddrNpi in the XML request?
A: SOURCE-ADDR-NPI (integer, 1) - Numbering plan indicator for source address. If not known, set to NULL (Unknown).

Q: What is destinationAddr in the XML request?
A: DESTINATION-ADDR (Variable, max 21, C Octet String) - Destination address of the short message, usually the recipient MS's directory number.

Q: What is destAddrTon in the XML request?
A: DEST-ADDR-TON (integer, 1) - Type of number for destination address.

Q: What is destAddrNpi in the XML request?
A: DEST-ADDR-NPI (integer, 1) - Numbering plan indicator for destination address.

Q: What is esmClass in the XML request?
A: ESM-CLASS (integer, 1) - Indicates message mode and message type.

0: Short message
64: Long message [UDHI flag]
Q: What is concatenatedSMesmClass in the XML request?
A: concatenatedSMesmClass (integer, 1) - Indicates message mode and message type, like ESM-CLASS.

0: Short message
64: Long message [UDHI flag]
Q: What is dataCoding in the XML request?
A: DATA-CODING (integer, 1) - Defines the encoding scheme of the short message user data.

0: English
8: Other language
Q: What is shortMessage in the XML request?
A: SHORT-MESSAGE (Variable, 0-254, Octet String) - Up to 254 octets of short message user data.

Q: What is messagePayload in the XML request?

A: MESSAGE-PAYLOAD (boolean) -

True: Sends a long message in a single packet, without multi-part.
False: Sends a long message in multi-part, with UDH.
Q: What is registeredDelivery in the XML request?
A: REGISTERED-DELIVERY (integer, 1) - Indicator for SMSC delivery receipt or SME acknowledgement.

0: No delivery receipt
1: Delivery receipt required
Q: What is protocolId in the XML request?
A: PROTOCOL-ID (integer, 1) - Protocol identifier.

64: Silent
0: Normal
Q: What is priorityFlag in the XML request?
A: PRIORITY-FLAG (integer, 1) - Designates the priority level of the message.

Q: What is replaceIfPresentFlag in the XML request?
A: REPLACE-IF-PRESENT-FLAG (integer, 1) - Flag indicating if the submitted message should replace an existing message.

Q: What is smDefaultMsgId in the XML request?
A: SM-DEFAULT-MSG-ID (integer, 1) - Indicates the short message to send from a list of predefined ("canned") short messages stored on the SMSC. If not using an SMSC canned message, set to NULL.

How does the acknowledgement work in the submit_sm API?
A: In submit_sm, acknowledgement is given when the request hits the ESME URL. After processing, the original response is given to the resp URL attached in the request API.

Q: Is there any acknowledgement in synch_submit_sm API?
A: No, in synch_submit_sm, the request and response are generated in the same cycle simultaneously, so no acknowledgement is given.

Q: What adapter can be used if a response is not needed in submit_sm API?

A: The HttpNoRespAdapter adapter can be used if no response is needed. It avoids the extra effort of sending the response to the resp URL attached to the request API packet.

Q: How does the response behave in synch_submit_sm?
A: In synch_submit_sm, the response is synchronized with the request, meaning both are generated simultaneously within the same cycle.

Q: What is the adapter used in synch_submit_sm for the request and response process?
A: The HttpAdapter is used in synch_submit_sm for handling the request and response process.

Q: What is the maximum number of octets that can be sent in a submit_sm request?
A: The maximum number of octets that can be sent in a submit_sm request is 254 octets, according to standard SMPP configuration.

Q: What are the methods to send larger user data sizes from ESME to SMSC?
A: To send larger user data sizes, there are two methods:

Multipart message with UDH (User Data Header).
Use of message payload with a flag indicating if the message is sent as multipart.
Q: What is a Multipart message with UDH?
A: A multipart message with UDH involves splitting the message into segments, each with a separate request. Each message part includes a UDH that indicates it is part of a multipart message. The parameters in the PDU body are esm_class (set to 0x40 to indicate UDH) and short_message (which contains the UDH information at the beginning of the message).

Q: What does the esm_class value of 0x40 signify in a multipart message?
A: The esm_class value of 0x40 (64 in decimal) indicates that there is a User Data Header (UDH) in the message.

Q: What are the components of a concatenated message UDH structure?
A: The UDH structure for concatenated messages consists of:

Length of UDH (5 bytes)
Indicator for concatenated message
Subheader length (3 bytes)

Message identification (a hexadecimal number matching the UDH Reference Number)
Number of pieces of the concatenated message
Sequence number for message concatenation
Q: What are the maximum characters per concatenated message for different encodings?
A: The maximum characters per concatenated message depend on the encoding:

153 characters for 7-bit encoding (e.g., Latin-1/9, GSM8)
134 characters for 8-bit encoding (Binary)
67 characters for 16-bit encoding (Unicode)
Q: How are the ESM-CLASS and concatenatedSMesmClass tags configured in the request XML for multipart messages?
A: In the request XML from MO Router to ESME, the ESM-CLASS and concatenatedSMesmClass tags should be set to 64 (<ESM-CLASS>64</ESM-CLASS> and <concatenatedSMesmClass>64</concatenatedSMesmClass>).

Q: How is message payload configured in the request XML for enabling multipart?
A: For enabling message payload, you need to configure the following tags in the request XML:

<ESM-CLASS>0</ESM-CLASS>
<concatenatedSMesmClass>0</concatenatedSMesmClass>
<MESSAGE-PAYLOAD>true</MESSAGE-PAYLOAD>
Q: What does the <MESSAGE-PAYLOAD>true</MESSAGE-PAYLOAD> tag indicate in the request XML?
A: The <MESSAGE-PAYLOAD>true</MESSAGE-PAYLOAD> tag indicates that the message should be sent as a multipart message, allowing longer message sizes.

Q: What is the purpose of multiple SMPP support in ESME?
A: ESME supports multiple SMPP accounts, allowing it to connect to two or more SMSC accounts. Each SMSC account needs to be configured separately in the smpp.xml file, and different bind modes can be configured for different accounts.

Q: How do you enable retry functionality in ESME?
A: To enable the retry functionality in ESME, you need to set the synchSubmitSm property to retrySubmitSm and the roundRobinSupport property to false. You also need to modify the retry table configuration in the database and adjust the web.xml file to set the Retry parameter to Y.

Q: What properties are configured for retry functionality in ESME?
A: The following properties are configured for retry functionality:

```
synchSubmitSm = retrySubmitSm
roundRobinSupport = false
sendretryresponse = true (optional, for retry response handling)
```

Additionally, the RETRY_TABLE contains columns such as REQUEST_ID, REQUEST, FIRST_ATTEMPTS, LAST_ATTEMPTS, NEXT_ATTEMPTS, ATTEMPTS_CNT, and STATUS.

Q: What is the structure of the RETRY_TABLE used for retry functionality?
A: The RETRY_TABLE includes the following columns:

REQUEST_ID: Unique identifier for the request.

REQUEST: The request object.

FIRST_ATTEMPTS: The date and time of the first attempt.

ATTEMPTS_CNT: The count of retry attempts.

LAST_ATTEMPTS: The date and time of the last attempt.

STATUS: The status of the request (Y or N).

NEXT_ATTEMPTS: The date and time of the next attempt.

How do you enable retry in the web.xml file?
A: To enable retry, change the param-value for the Retry parameter in web.xml from N to Y:

```
<init-param>
  <param-name>Retry</param-name>
  <param-value>Y</param-value>
</init-param>
```

Q: How is the database connection configured for retry functionality in standalone.xml?
A: The database connection is configured in the <datasource> section of standalone.xml, where you define the connection URL, driver, pool size, and other properties for the database (Oracle or MySQL).

For example, for Oracle:

```
<datasource jndi-name="java:/OracleDS" pool-name="OracleDS" enabled="true">
  <connection-url>jdbc:oracle:thin:@10.0.14.44:1521/bluesky</connection-url>
  <driver>oracle</driver>
  <pool>
    <min-pool-size>15</min-pool-size>
    <max-pool-size>30</max-pool-size>
  </pool>
  <security>
    <user-name>NGWUSER</user-name>
    <password>NGWUSER</password>
  </security>
</datasource>
```

Q: How is the SMPP.XML file configured for retry?
A: In smpp.xml, the tps (transactions per second) should be set to the desired rate, for example:

```
<tps>1</tps>
```

Q: How is the MySQL connection configured in standalone.xml when using IPv6?
A: The MySQL connection using IPv6 is configured as follows:

```
<connection-url>jdbc:mysql://address=(protocol=tcp)(host=2406:7400:f5:901::a)(port=3306)/esme?zeroDateTimeBehavior=convertToNull</connection-url>
```

Q: What is the format for testing the retry functionality with a dummy packet?
A: A dummy packet for testing the retry functionality is structured as follows:

```
<REQ>
  <FEATURE>retry</FEATURE>
  <TIME-STAMP>29032021184543</TIME-STAMP>

<RESPONSE-URL>http://localhost:8080/Sampleproject-0.0.1-SNAPSHOT/AppTest1</RESPONSE-URL>
  <PARAMETERS>
    <REQ-TRANSACTION-ID>335</REQ-TRANSACTION-ID>
    <SMSC-ID>airtel1</SMSC-ID>
    <SUBMIT-SM>
      <SOURCE-ADDR>555</SOURCE-ADDR>
      <DESTINATION-ADDR>168425861</DESTINATION-ADDR>
      <SHORT-MESSAGE>NkQgTEFCIFRFU1Q=</SHORT-MESSAGE>
    </SUBMIT-SM>
  </PARAMETERS>
</REQ>
```

Note that the <SMSC-ID> is intentionally given as incorrect (e.g., airtel1) to trigger retry attempts.

Q: What happens when SMSC transfers messages longer than 160 characters to ESME?
A: When SMSC transfers messages longer than 160 characters to ESME, the message is divided into multiple part-messages with corresponding UDH headers. These headers are reserved for message concatenation, which is handled by the Mo Router or equivalent module.

Q: Why are UDH headers present in part-messages?
A: UDH headers are present in part-messages to enable the Mo Router or equivalent module to perform message concatenation. They are essential for reconstructing the original message.

Q: How does the Mo Router use the UDH headers?
A: The Mo Router decodes the base64-encoded <SMS-BYTE> tag, which contains the UDH headers. These headers are used for message concatenation, allowing the Mo Router to combine part-messages into a single message.

Q: How are messages sent from SMSC to ESME encoded?
A: Messages sent from SMSC to ESME are encoded in byte format and transferred with base64 encoding in the <SMS-BYTE> tag of the request XML.

Q: Does the ESME handle message concatenation?
A: No, ESME does not handle message concatenation. The Mo Router handles concatenation using the UDH headers.

Q: What is the purpose of the <messageformat> tag in the provided XML example?

A: The <messageformat> tag defines the format for the message, including placeholders for various message details such as time-stamp, transaction ID, MSISDN, and the SMS content. It is used to structure the message data in a predefined format.

Q: What does the <logs> section in Smpp.xml configure?
A: The <logs> section in the Smpp.xml file configures the log format pattern for sending logs to ELK. It includes placeholders for various parameters like transaction IDs, MSISDN, status codes, and other message details.

Q: What does ELK stand for in the context of logging?
A: ELK stands for Elasticsearch, Logstash, and Kibana. It is a stack used for processing, storing, and visualizing logs and other event data.

Q: How does Logstash interact with Elasticsearch in the ELK stack?
A: Logstash extracts logging data or events from input sources, processes them, and stores them in Elasticsearch for querying and visualization.

Q: What is the role of Kibana in the ELK stack?
A: Kibana is a web interface that accesses the logging data stored in Elasticsearch and visualizes it for easy analysis.

Q: How can the log format be customized in the ELK configuration?
A: The log format can be customized in the <logs> section of Smpp.xml, where placeholders for various parameters (e.g., time-stamp, transaction ID, MSISDN) are defined. These placeholders can be adjusted based on client requirements.

Q: What does the log4j appender in log4j.xml support?
A: The log4j appender in log4j.xml is written to support ELK logging, enabling logs to be sent to the ELK stack for processing and visualization.

Q: What does the Data Coding value "00000000" represent in ESME?
A: The Data Coding value "00000000" represents the SMSC Default Alphabet.

Q: What is the meaning of the Data Coding value "00000001"?
A: The Data Coding value "00000001" refers to IA5 (CCITT T.50) or ASCII (ANSI X3.4).

Q: What does "00000010" in the Data Coding field indicate?
A: "00000010" in the Data Coding field indicates that the octet is unspecified (8-bit binary).

Q: What encoding is used for "00000011" in Data Coding?
A: "00000011" in Data Coding refers to Latin 1 (ISO-8859-1).

Q: What is the meaning of "00000100" in the Data Coding field?
A: "00000100" in the Data Coding field represents an octet unspecified (8-bit binary) with a specific meaning that is not defined.

Q: What encoding is associated with the Data Coding value "00000101"?
A: The Data Coding value "00000101" corresponds to JIS (X 0208-1990) encoding.

Q: What does "00000110" in Data Coding stand for?
A: "00000110" in Data Coding represents Cyrillic (ISO-8859-5) encoding.

Q: What does the Data Coding value "00000111" refer to?
A: The Data Coding value "00000111" refers to Latin/Hebrew (ISO-8859-8) encoding.

Q: What encoding is indicated by the Data Coding value "00001000"?
A: "00001000" in Data Coding represents UCS2 (ISO/IEC-10646) encoding.

Q: What does "00001001" in the Data Coding field represent?
A: "00001001" in Data Coding represents Pictogram Encoding.

Q: What is indicated by the Data Coding value "00001010"?
A: "00001010" in Data Coding refers to ISO-2022-JP (Music Codes).

Q: What does "00001011" represent in the Data Coding field?
A: "00001011" in Data Coding is reserved.

Q: What is the meaning of "00001100" in Data Coding?
A: "00001100" in Data Coding is reserved.

Q: What does the Data Coding value "00001101" refer to?
A: "00001101" in Data Coding corresponds to Extended Kanji JIS (X 0212-1990) encoding.

Q: What does "00001110" in Data Coding represent?
A: "00001110" in Data Coding corresponds to KS C 5601 encoding.

Q: What does the Data Coding value "00001111" indicate?
A: "00001111" in Data Coding is reserved.

Q: What does the Data Coding value "10111111" represent?
A: "10111111" in Data Coding is reserved.

Q: What is the meaning of the Data Coding values starting with "1100xxxx"?
A: The Data Coding values starting with "1100xxxx" represent GSM MWI control, as defined in [GSM 03.38].

Q: What does "1101xxxx" in Data Coding refer to?
A: "1101xxxx" in Data Coding refers to GSM MWI control, as defined in [GSM 03.38].

Q: What is the meaning of Data Coding values starting with "1110xxxx"?
A: The Data Coding values starting with "1110xxxx" are reserved.

Q: What do the Data Coding values starting with "1111xxxx" represent?
A: The Data Coding values starting with "1111xxxx" are for GSM message class control, as defined in [GSM 03.381].

Q: What Data Coding Scheme (DCS) value is used for English language messages in ESME?
A: The Data Coding Scheme (DCS) value used for English language messages is "0", which corresponds to the SMSC Default Alphabet.

Q: What does the UDHI indicator indicate in the trace of a long multi-part message?
A: The UDHI (User Data Header Indicator) indicator, when set to 1, indicates that the message is a multi-part message.

Q: What are the two common scenarios for supporting foreign languages in ESME?
A: The two common scenarios for supporting foreign languages are:

Foreign language in English script (e.g., Indonesian in English script).

Foreign language in its own script (e.g., Arabic in Arabic script).
Q: How is foreign language in English script handled in ESME?
A: Foreign language in English script (e.g., Indonesian in English script) can be handled by using DCS zero, the same configuration as for English language messages.

Q: What DCS value is used when sending foreign language messages in their own script, such as Arabic?
A: For foreign languages in their own script (e.g., Arabic), DCS value "8" is used, which corresponds to UCS2 encoding.

Q: How does the MO Router handle foreign language scripts like Arabic?
A: The MO Router processes the Decimal NCR (National Character References) equivalent of the required language script and converts it into a base-64 encoded string. This string is then passed to ESME, which converts it into the corresponding SMPP equivalent (hex string) and sends it to SMSC.

Q: Can foreign language scripts be directly placed in the message templates table at the MO Router?
A: No, foreign language scripts cannot be directly placed in the message templates table at the MO Router. Only English script works like that. For foreign languages, the message needs to be converted into Decimal NCR equivalents.

Q: What encoding is typically used for Arabic language messages in ESME?
A: Arabic language messages are typically encoded using UCS2 (DCS value 8) to support the Arabic script.

Q: What issue arises when sending French characters using DCS value "3"?
A: When DCS value "3" (Latin 1/ISO-8859-1 encoding) is used for French language messages, some characters are not available in GSM 7-bit encoding. These characters appear on a grey background.

Q: How can the issue of missing French characters be addressed?
A: To address the issue of missing French characters, DCS value "8" (UCS2 encoding) should be used, similar to how Arabic language messages are handled.

Q: What changes were made to the encoding for French language messages in the MO Router?

A: Initially, the MO Router used UTF-16 encoding for French language messages, which took two bytes per character. Later, it switched to ISO-8859-1 encoding, which uses a single byte per character, allowing for more GSM 7-bit characters.

Q: Does ESME handle the conversion of Decimal NCR into base-64 encoded short messages?
A: No, ESME does not handle the conversion of Decimal NCR into base-64 encoded short messages. This logic is handled by the MO Router.

What is the error code for an unknown feature or illegal feature?

Answer: -1001, Unknown feature or Illegal feature
Which error code corresponds to overload rejection?

Answer: -1002, Overload rejection
What does error code -1003 indicate?

Answer: Internal server error (check feature_id)
What error code is associated with an unknown operation or exception?

Answer: -1004, Unknown Operation or Exception
Which error code is for an illegal request message?

Answer: 1000, Illegal request Message
What error code signifies a mandatory field is missing?

Answer: 1005, Mandatory field missing
Which error code represents an illegal data type?

Answer: 1006, Illegal data type
What error code occurs when the username or password does not match?

Answer: 1007, Username or password is not match
What does error code 1008 signify?

Answer: Other Error, i.e. Not able to send response
Which error code is related to an unknown CommandId?

Answer: 1009, Unknown CommandId
What error code indicates an execute exception?

Answer: 1010, Execute Exception
Which error code corresponds to CMD_FIELD_MISSING_ERRORCODE?

Answer: 1011, CMD_FIELD_MISSING_ERRORCODE
What error code indicates an unknown CommandId error?

Answer: 1012, UNKNOWN_COMMANDID_ERRORCODE
Which error code is related to an unknown SMSC error?

Answer: 1013, UNKNOWN_SMSC_ERRORCODE

What does error code 1014 signify?

Answer: NOT_BOUND_WITH_SMSC_ERRORCODE
Which error code corresponds to an already bound with SMSC error?

Answer: 1015, ALREADY_BOUND_WITH_SMSC_ERRORCODE
What error code is associated with an IO exception?

Answer: 1016, IOEXCEPTION_ERRORCODE
Which error code represents a negative response?

Answer: 1017, NEGATIVE_RESPONSE_ERRORCODE
What error code signifies a response timeout?

Answer: 1018, RESPONSE_TIMEOUT_ERRORCODE
Which error code corresponds to an unsuccessful delivery of an SME?

Answer: 1019, UNSUC_DELIVERY_SME_ERRORCODE
What does error code 1020 signify?

Answer: THROTTLING_LIMIT_ERRORCODE

What is the error code for "No Error"?

Answer: 0x00000000, No Error
Which error code indicates an invalid message length?

Answer: 0x00000001, Message Length is invalid
What error code corresponds to an invalid command length?

Answer: 0x00000002, Command Length is invalid
Which error code represents an invalid command ID?

Answer: 0x00000003, Invalid Command ID
What does error code 0x00000004 signify?

Answer: Incorrect BIND Status for given command
What error code represents "ESME Already in Bound State"?

Answer: 0x00000005, ESME Already in Bound State
Which error code corresponds to an invalid priority flag?

Answer: 0x00000006, Invalid Priority Flag
What error code indicates an invalid registered delivery flag?

Answer: 0x00000007, Invalid Registered Delivery Flag
What does error code 0x00000008 signify?

Answer: System Error
Which error code is reserved for future use?

Answer: 0x00000009, Reserved
What error code corresponds to an invalid source address?

Answer: 0x0000000A, Invalid Source Address
What does error code 0x0000000B indicate?

Answer: Invalid Destination Address
Which error code indicates an invalid message ID?

Answer: 0x0000000C, Message ID is invalid
What error code is for a bind failure?

Answer: 0x0000000D, Bind Failed
Which error code represents an invalid password?

Answer: 0x0000000E, Invalid Password
What does error code 0x0000000F signify?

Answer: Invalid System ID
Which error code is reserved for future use?

Answer: 0x00000010, Reserved
What error code represents a cancel SM failure?

Answer: 0x00000011, Cancel SM Failed
Which error code corresponds to a replace SM failure?

Answer: 0x00000013, Replace SM Failed
What does error code 0x00000014 indicate?

Answer: Message Queue Full
Which error code represents an invalid service type?

Answer: 0x00000015, Invalid Service Type
What error code corresponds to an invalid number of destinations?

Answer: 0x00000033, Invalid number of destinations
Which error code indicates an invalid distribution list name?

Answer: 0x00000034, Invalid Distribution List name
What error code is associated with an invalid destination flag (submit_multi)?

Answer: 0x00000040, Destination flag is invalid (submit_multi)
What does error code 0x00000042 signify?

Answer: Invalid 'submit with replace' request (e.g., submit_sm with replace_if_present_flag set)
Which error code corresponds to invalid esm_class field data?

Answer: 0x00000043, Invalid esm_class field data
What does error code 0x00000044 indicate?

Answer: Cannot Submit to Distribution List
What error code corresponds to throttling error (ESME has exceeded allowed message limits)?

Answer: 0x00000058, Throttling error
Which error code represents an invalid scheduled delivery time?

Answer: 0x00000061, Invalid Scheduled Delivery Time
What does error code 0x00000062 signify?

Answer: Invalid message validity period (Expiry time)
Which error code represents a predefined message invalid or not found?

Answer: 0x00000063, Predefined Message Invalid or Not Found
What error code is associated with a temporary application error from the receiver?

Answer: 0x00000064, ESME Receiver Temporary App Error Code
Which error code corresponds to a permanent application error from the receiver?

Answer: 0x00000065, ESME Receiver Permanent App Error Code
What does error code 0x00000067 indicate?

Answer: query_sm request failed
Which error code represents a delivery failure?

Answer: 0x000000FE, Delivery Failure (used for data_sm_resp)
What does error code 0x000000FF signify?

Answer: Unknown Error
What error code is reserved for SMPP extension?

Answer: 0x000003FF, Reserved for SMPP extension
Which error code is reserved for SMSC vendor-specific errors?

Answer: 0x000004FF, Reserved for SMSC vendor-specific errors
What does error code 0xFFFFFFFF represent?

Answer: Reserved Message State

Question: What is the IP address used to connect to the SMSC?
Answer: The IP address used for the SMSC connection is specified in the <ip-address> tag.
Example from config:

```
<ip-address>127.0.0.1</ip-address>
```

Question: Which port is used for the SMPP connection?
Answer: The SMPP client connects to the SMSC on port 2778, as defined in the <port> tag.
Example from config:

```
<port>2778</port>
```

Question: What is the system ID used for authentication?
Answer: The system ID is the client identifier used for authenticating with the SMSC.
Example from config:

```
<system-id>test1</system-id>
```

Question: What is the password used for the SMPP bind request?
Answer: The password for SMPP binding is specified in the <password> tag.
Example from config:

```
<password>test1</password>
```

Question: What type of protocol is defined in the configuration?
Answer: The protocol is defined as “smpp”, which stands for Short Message Peer-to-Peer.
Example from config:

```
<smsc id=“airtel” protocol=“smpp”>
```

Question: What does <bind-mode>trx</bind-mode> mean?
Answer: The bind mode “trx” indicates that the client is using the SMPP transceiver mode, allowing it to both send and receive messages.
Example from config:

```
<bind-mode>trx</bind-mode>
```

Question: What does <receive-timeout>5</receive-timeout> represent?
Answer: It means the client waits for up to 5 seconds when trying to receive a message from the SMSC in synchronous mode.
Example from config:

```
<receive-timeout>5</receive-timeout>
```

Question: What is the purpose of <enquireLinkTimer>?
Answer: This defines the interval in seconds at which the client sends an enquire link to the SMSC to keep the connection alive.
Example from config:

```
<enquireLinkTimer>5</enquireLinkTimer>
```

Question: What does <pdu-processors> define?

Answer: It specifies the number of threads that can simultaneously process SMPP requests.
Example from config:

```
<pdu-processors>3</pdu-processors>
```

Question: What does <pdu-maxQueueSize> do?
Answer: It defines the size of the queue for PDU (Protocol Data Unit) processors. A value of -1 means the queue size is infinite.
Example from config:

```
<pdu-maxQueueSize>-1</pdu-maxQueueSize>
```

Question: What does <addr-ton> specify?
Answer: It sets the Type of Number (TON) for the address. A value of 0 typically means

it's unknown or international depending on the implementation.

Example from config:

```
<addr-ton>0</addr-ton>
```

Question: What is the meaning of <addr-npi> in SMPP config?

Answer: It defines the Numbering Plan Indicator (NPI). A value of 0 usually means unknown.

Example from config:

```
<addr-npi>0</addr-npi>
```

Question: What address range will this SMPP client serve?

Answer: The <addr-range> tag specifies the address range. "NULL" means it will accept all addresses.

Example from config:

```
<addr-range>NULL</addr-range>
```

Question: What is <service-type> used for in this config?

Answer: It defines the type of service. If set to NULL, it means no specific service type is used.

Example from config:

```
<service-type>NULL</service-type>
```

Question: What does <sourceAddrTon> represent?
Answer: It sets the Type of Number for the source address. A value of 0 indicates unknown.
Example from config:

```
<sourceAddrTon>0</sourceAddrTon>
```

Question: What is <sourceAddrNpi> used for?
Answer: It defines the Numbering Plan Indicator for the source address. Value 0 means unknown.
Example from config:

```
<sourceAddrNpi>0</sourceAddrNpi>
```

Question: What does <destAddrTon> specify?
Answer: It indicates the Type of Number for the destination address. 0 means unknown.
Example from config:

```
<destAddrTon>0</destAddrTon>
```

Question: What is the value of <destAddrNpi> and what does it mean?
Answer: It's 0, which means the numbering plan is unknown for the destination address.
Example from config:

```
<destAddrNpi>0</destAddrNpi>
```

Question: What does <esmClass> represent?
Answer: The esmClass defines the message mode and type. A value of 0 usually means a default mode message.
Example from config:

```
<esmClass>0</esmClass>
```

Question: What is <concatenatedSMesmClass> used for?
Answer: It is used to set ESM class for concatenated SMS messages. A value of 0 means it's not being used.
Example from config:

```
<concatenatedSMesmClass>0</concatenatedSMesmClass>
```

Question: What does <protocolId> define?
Answer: The protocol ID field defines protocol-specific values. A value of 0 typically means the default protocol.
Example from config:

```
<protocolId>0</protocolId>
```

Question: What is <priorityFlag> used for?
Answer: It sets the priority of the message. A value of 1 means the message has high priority.
Example from config:

```
<priorityFlag>1</priorityFlag>
```

Question: What does <registeredDelivery> specify?
Answer: It controls delivery receipt behavior. A value of 0 means no delivery receipt is requested.
Example from config:

```
<registeredDelivery>0</registeredDelivery>
```

Question: What is <replaceIfPresentFlag> used for?
Answer: If a message with the same ID exists, this flag decides whether to replace it. 0 means do not replace.
Example from config:

```
<replaceIfPresentFlag>0</replaceIfPresentFlag>
```

Question: What does <smDefaultMsgId> mean?
Answer: It sets the default message ID for pre-defined messages. 0 means no default message is set.
Example from config:

```
<smDefaultMsgId>0</smDefaultMsgId>
```

Question: What does <allowedNumSeries> represent?
Answer: It defines a regular expression pattern that allows only numbers matching this

format. The example allows 9-digit numbers ending with a 3.
Example from config:

```
<allowedNumSeries>[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]3</allowedNumSeries>
```

Question: What is <messageReceiverListenerImpl> used for?
Answer: It defines the Java class that processes incoming messages from SMSC.
Example from config:

```
<messageReceiverListenerImpl>com.sixdee.imp.receiver.MessageReceiverListenerImpl</messageReceiverListenerImpl>
```

Question: What does <dcs> stand for in SMPP config?
Answer: DCS stands for Data Coding Scheme. A value of 0 typically means GSM 7-bit default encoding.
Example from config:

```
<dcs>0</dcs>
```

Question: What is <inByte> in the SMPP config?
Answer: It's a custom or extended tag. Here, 0 might represent a default or disabled value.
Example from config:

```
<inByte>0</inByte>
```

Question: What does <tps> mean and why is it set to -1?
Answer: TPS stands for Transactions Per Second. A value of -1 indicates no limit is applied on message throughput.
Example from config:

```
<tps>-1</tps>
```

Question: What does <max_connection> define in the connector?
Answer: It defines the number of simultaneous threads that can send messages through HTTP to the NG/third-party system.
Example:

```
<max_connection>5</max_connection>
```

Question: What transport protocol is used for sending messages?
Answer: The protocol is HTTP, as specified in the <transportprotocol> tag.
Example:

```
<transportprotocol>http</transportprotocol>
```

Question: What HTTP method is used to send messages in the connector?
Answer: The method is POST.
Example:

```
<transportmethod>post</transportmethod>
```

Question: What is the primary transport URL for sending messages?
Answer: It's the URL defined under <transporturl1>.
Example:

```
<transporturl1>http://10.0.0.69:8076/UMS/HttpAdapter</transporturl1>
```

Question: What is the secondary transport URL in case the primary fails?
Answer: Defined in <transporturl2>, it's a fallback endpoint.
Example:

```
<transporturl2>http://10.0.0.69:9650/ProfileManager_Ums/HttpAdapter</transporturl2>
```

Question: What is the structure of the message format used in this connector?
Answer: The message is formatted as XML with placeholders for timestamp, transaction ID, SMPP ID, ESM class, sender/receiver info, DCS, and message metadata.
Example (simplified):

```
<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>
  <MESSAGE>{17}</MESSAGE>
</XML>]]>
</messageformat>
```

Q&A with Examples for <deliveryconnector> Section

Question: What is the role of <deliveryconnector>?
Answer: It is used to send delivery receipts (DLRs) to the NG or third-party system over HTTP.

Question: How many concurrent delivery receipt threads can be handled?
Answer: As per the <max_connection> tag, up to 5 concurrent connections.
Example:

```
<max_connection>5</max_connection>
```

Question: Which protocol and method are used for delivery receipts?
Answer: The delivery connector uses HTTP and the POST method.
Example:

```
<transportprotocol>http</transportprotocol>
<transportmethod>post</transportmethod>
```

Question: What is the main delivery receipt transport URL?
Answer: Defined in <transporturl1>.
Example:

```
<transporturl1>http://10.0.0.65:8022/MCA/HttpAdapter</transporturl1>
```

Question: What is the backup URL for delivery receipts?
Answer: Defined in <transporturl2>.
Example:

```
<transporturl2>http://10.0.0.69:9650/ProfileManager_Ums/HttpAdapter</transporturl2>
```

Question: What is the format of a delivery receipt message?
Answer: It's an XML structure that includes delivery metadata such as time, transaction ID, SMPP ID, command ID, sender, receiver, and message ID.
Example:

```
<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>
  <OA>{5}</OA>
  <DA>{6}</DA>
  <DCS>{7}</DCS>
  <SMS>{8}</SMS>
  <MESSAGE-ID>{10}</MESSAGE-ID>
</REQ>]]>
</messageformat>
```

Question: What are the placeholders like {0}, {1}, {8} used for in messageformat?
Answer: These placeholders are dynamically replaced with actual values at runtime such as timestamps, IDs, sender/receiver numbers, etc.

Question: What happens if transporturl1 is down?
Answer: The system can fall back to transporturl2 if implemented in logic to ensure message delivery.

Section: <optionalparameterstag>

Question: What is <optionalparameterstag> used for?
Answer: It defines TLVs (Tag-Length-Value) optional parameters that may be sent along with SMPP messages.

Question: What does tag 5126 represent in this configuration?
Answer: The tag 5126 is associated with the parameter OptionalParam_OA.

Example:

```
<tlv1>
  <tag>5126</tag>
  <name>OptionalParam_OA</name>
</tlv1>
```

Section: <Heartbeats>

Question: What is the role of the <Heartbeats> configuration block?
Answer: It defines settings for a heartbeat process that monitors the application's health.

Question: What does <HeartBeatPath> specify?
Answer: It provides the path to the heartbeat executable (JHeartBeat.exe).

Question: How frequently is the heartbeat sent?
Answer: Every 1000 milliseconds (1 second), as defined in <HeartBeatsInterval>.

Example:

```
<HeartBeatPath>/home/smsuser/ProcessManager/JHeartBeat.exe</HeartBeatPath>
<AppManagerID>208</AppManagerID>
```

```
<AppId>1</AppId>
<HeartBeatsInterval>1000</HeartBeatsInterval>
```

Section: <cdr>

Question: What does the <cdr> section define?
Answer: It sets the log format for Call Detail Records (CDRs) for SMS requests and responses.

Question: What kind of information does it log?
Answer: It logs request and response details like command ID, message ID, sequence number, address types, encoding, and more.

Question: How are values inserted into the CDR format?
Answer: By using placeholders like {0}, {1}, etc., which get replaced by real values at runtime.

Example (snippet):

```
Header = [Smsc Id = {0}, Log time = {1}, Request Id = {2}, ...]
REQ = [Status Code = {5}, command_id(hex) = {6}, ...]
RESP = [command_id = {26}, command_status = {27}, sequence_number = {28}, ...]
```

Section: <logs>

Question: What is the purpose of the <logs> block?
Answer: It defines how message logs are formatted for system-level or transaction-level logging.

Question: What details are included in this log pattern?
Answer: Timestamp, MSISDN, transaction and client IDs, message content (hex & plain), command status, service type, and more.

Question: What format is used for timestamps in logs?
Answer: The timestamp uses this pattern: yyyy-MM-dd'T'HH:mm:ss XXX.

Example (log pattern snippet):

```
TimeStamp::{1,date,yyyy-MM-dd'T'HH:mm:ss XXX}||ClientTransactionId::{3}||...
ShortMessageHex::{25}||ShortMessage::{32}
```

Question: What is the smscId used for?
Answer: It identifies the SMSC connection uniquely, e.g., airtel.

Question: What does protocol: smpp mean?
Answer: It indicates the use of SMPP (Short Message Peer-to-Peer) protocol for messaging.

Question: What is the significance of ip-address and port?
Answer: These specify where the SMPP client will connect to reach the SMSC.
Example:

```
ip-address: 10.0.0.75
port: 2778
```

Section: system-id, password, system-type

Question: What is system-id used for in SMPP configuration?
Answer: It is the client login ID required to bind with the SMSC.
Example: system-id: test1

Question: What does system-type: TR represent?
Answer: It typically indicates the service type or usage (e.g., “TR” for test or telecom route).

Section: bind-mode

Question: What are valid bind-mode options and what does BIND_TRX mean?
Answer: Valid modes: BIND_TX, BIND_RX, BIND_TRX

- BIND_TRX = transceiver mode (send and receive on same session).

Section: receive-timeout and enquireLinkTimer

Question: What does receive-timeout define?
Answer: Time (in ms) to wait for an incoming SMPP message.

Question: What is the purpose of enquireLinkTimer?
Answer: It's the interval (in ms) for sending link check (heartbeat) messages.

Example:

```
receive-timeout: 1000
enquireLinkTimer: 1000
```

Section: pduProcessorDegree, pduMaxQueueSize

Question: What does pduProcessorDegree represent?
Answer: Number of threads to process SMPP PDUs simultaneously.
Example: pduProcessorDegree: 3

Question: What does pduMaxQueueSize: -1 mean?
Answer: A value of -1 means the queue size is infinite.

Section: Address TON/NPI and Ranges

Question: What are addr-ton, addr-npi, sourceAddrTon, destAddrTon, etc.?
Answer: These define the Type of Number (TON) and Numbering Plan Indicator (NPI) for addresses.

Example:

yaml

CopyEdit

```
sourceAddrTon: 4
sourceAddrNpi: 4
destAddrTon: 4
destAddrNpi: 4
```

Section: SMPP Config

Question: What is esmClass used for?
Answer: It indicates message mode and type, such as normal message, delivery receipt, etc.

Question: What does allowedNumSeries: ..* mean?
Answer: It allows any numeric pattern for sender numbers (regex style).

Question: What is dcs: 0 used for?
Answer: Stands for Data Coding Scheme. 0 usually means default GSM 7-bit encoding.

Section: Message Listener, TPS, etc.

Question: What is the messageReceiverListenerImpl for?
Answer: It refers to the Java class that listens for incoming messages.

Question: What does tps: -1 represent?
Answer: TPS (Transactions per Second); -1 indicates no throttling.

yaml
CopyEdit

```
smsc:
  - smscId: airtel
    protocol: smpp
    ip-address: 10.0.0.75
    port: 2778
    system-id: test1
    password: test1
    system-type: TR
    bind-mode: BIND_TRX
    receive-timeout: 1000
    enquireLinkTimer: 1000
    pduProcessorDegree: 3
    pduMaxQueueSize: -1
    sourceAddrTon: 4
    allowedNumSeries: ..*
    messageReceiverListenerImpl: com.sixdee.imp.receiver.MessageReceiverListenerImpl
```

Section: backendThreshold, liveThreshold

Question: What does backendThreshold represent?
Answer: It defines the number of backend retry attempts before failing or escalating.
Example: backendThreshold: 2 – Retry 2 times before considering backend failed.

Question: What does liveThreshold mean?
Answer: It's the number of retry attempts in the "live" or high-priority retry pool.
Example: liveThreshold: 5 – Max 5 retries in live priority.

Question: What is maxAttempts used for?
Answer: Sets the total number of retry attempts allowed per message.
Example: maxAttempts: 3 – A message will be retried 3 times max.

Question: What is maxInPoolLimit?
Answer: Maximum number of retry messages allowed in the retry pool at once.
Example: maxInPoolLimit: 5 – Only 5 retry messages can sit in queue at a time.

Question: What is maxTpsPoolLimit?
Answer: The max number of TPS (transactions per second) allowed from retry pool.
Example: maxTpsPoolLimit: 4

Question: What does msgExpiryIntervalAfterFirstAttempts define?

Answer: It’s the time (in ms) after the first attempt that a message remains valid for retries.

Example: 86400000 ms = 24 hours

Question: What is the retryInterval?

Answer: Delay (in milliseconds) between two retry attempts for the same message.

Example: retryInterval: 500000 = 500 seconds = ~8.3 minutes

Question: What does roundRobinSupport: false mean?

Answer: It disables round-robin retry logic between multiple SMSC/hosts.

Question: What does tps: 15 represent?

Answer: Maximum number of retry TPS (transactions per second) allowed overall.

Example: 15 retry messages can be sent per second at most.

```
retry:
  config:
    backendThreshold: 2
    liveThreshold: 5
    maxAttempts: 3
    maxInPoolLimit: 5
    maxTpsPoolLimit: 4
    msgExpiryIntervalAfterFirstAttempts: 86400000
    retryInterval: 500000
    roundRobinSupport: false
    synchSubmitSm: retrySubmitSm
    tps: 15
```