Open Service Access (OSA);
Parlay X web services;
Part 4: Short messaging
<Remarks>

1. Application level of English description
   Application level : E2
   English description is included in the text and figures of main body, annexes and appendices.

2. Relationship with international recommendations and standards
   This standard is standardized based on the Technical Specification 29.199-04(Version 6.1.0) approved by 3GPP.

3. Departures from international recommendations
   Changes to original standard
   Standards referred to in the original standard, which are replaced by TTC standards.
   Standards referred to in the original standard should be replaced by derived TTC standards.

4. IPR
   Status of “Confirmation of IPR Licensing Condition” submitted is provided in the TTC web site.

5. Working Group
   3GPP Working Group
Contents

Foreword....................................................................................................................................................... 5
Introduction ..................................................................................................................................................... 5
1 Scope ......................................................................................................................................................... 6
2 References .................................................................................................................................................. 6
3 Definitions and abbreviations .................................................................................................................. 7
3.1 Definitions .............................................................................................................................................. 7
3.2 Abbreviations ..................................................................................................................................... 7
4 Detailed service description .................................................................................................................... 7
5 Namespaces .............................................................................................................................................. 9
6 Sequence Diagrams ............................................................................................................................... 9
6.1 Send SMS and report status ................................................................................................................ 9
7 XML Schema data type definition .......................................................................................................... 10
7.1 DeliveryStatus enumeration .............................................................................................................. 10
7.2 SmsFormat enumeration .................................................................................................................... 10
7.3 DeliveryInformation structure ....................................................................................................... 11
7.4 SmsMessage structure ....................................................................................................................... 11
8 Web Service interface definition .......................................................................................................... 11
8.1 Interface: SendSms .............................................................................................................................. 11
8.1.1 Operation: SendSms ......................................................................................................................... 11
8.1.1.1 Input message: SendSmsRequest ............................................................................................... 12
8.1.1.2 Output message: SendSmsResponse ......................................................................................... 12
8.1.1.3 Referenced faults ....................................................................................................................... 12
8.1.2 Operation: SendSmsLogo ............................................................................................................... 12
8.1.2.1 Input message: SendSmsLogoRequest ...................................................................................... 13
8.1.2.2 Output message: SendSmsLogoResponse ............................................................................... 13
8.1.2.3 Referenced faults ....................................................................................................................... 13
8.1.3 Operation: SendSmsRingtone ........................................................................................................ 13
8.1.3.1 Input message: SendSmsRingtoneRequest .............................................................................. 14
8.1.3.2 Output message: SendSmsRingtoneResponse ........................................................................... 14
8.1.3.3 Referenced faults ....................................................................................................................... 14
8.1.4 Operation: GetSmsDeliveryStatus .............................................................................................. 15
8.1.4.1 Input message: GetSmsDeliveryStatusRequest ...................................................................... 15
8.1.4.2 Output message: GetSmsDeliveryStatusResponse ................................................................... 15
8.1.4.3 Referenced faults ....................................................................................................................... 15
8.2 Interface: SmsNotification ................................................................................................................. 15
8.2.1 Operation: NotifySmsNotification ............................................................................................... 15
8.2.1.1 Input message: NotifySmsNotificationRequest ...................................................................... 17
8.2.1.2 Output message: NotifySmsNotificationResponse ................................................................. 17
8.2.1.3 Referenced faults ....................................................................................................................... 17
8.3 Interface: ReceiveSms ......................................................................................................................... 17
8.3.1 Operation: GetReceivedSms ........................................................................................................... 17
8.3.1.1 Input message: GetReceivedSmsRequest ............................................................................... 17
8.3.1.2 Output message: GetReceivedSmsResponse ........................................................................... 17
8.3.1.3 Referenced faults ....................................................................................................................... 17
8.4 Interface: SmsNotificationManager ............................................................................................... 17
8.4.1 Operation: StartSmsNotificationManager ................................................................................... 17
8.4.1.1 Input message: StartSmsNotificationManagerRequest ............................................................ 18
8.4.1.2 Output message: StartSmsNotificationResponse................................................................. 18
8.4.1.3 Referenced Faults .............................................................................................................. 18
8.4.2 Operation: StopSmsNotification .......................................................................................... 18
8.4.2.1 Input message: StopSmsNotificationRequest................................................................. 18
8.4.2.2 Output message: StopSmsNotificationResponse........................................................... 18
8.4.2.3 Referenced Faults .............................................................................................................. 18

9 Fault definitions ..................................................................................................................... 19
  9.1 ServiceException .................................................................................................................. 19
  9.1.1 SVC0280: Message too long............................................................................................. 19
  9.1.2 SVC0281: Unrecognized data format ........................................................................... 19
  9.1.3 SVC0282: Overlapping Criteria ..................................................................................... 19
  9.1.4 SVC0283: Delivery Receipt Notification not supported............................................... 19

10 Service policies ..................................................................................................................... 19

Annex A (normative): WSDL for Short Messaging .................................................................... 20
Annex B (informative): Change history .................................................................................... 21
Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

3GPP acknowledges the contribution of the Parlay X Web Services specifications from The Parlay Group. The Parlay Group is pleased to see 3GPP acknowledge and publish the present document, and the Parlay Group looks forward to working with the 3GPP community to improve future versions of the present document.

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:
  1 presented to TSG for information;
  2 presented to TSG for approval;
  3 or greater indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

Introduction

The present document is part 4 of a multi-part deliverable covering the 3rd Generation Partnership Project; Technical Specification Group Core Network; Open Service Access (OSA); Parlay X Web Services, as identified below:

Part 1: "Common";
Part 2: "Third party call";
Part 3: "Call Notification";
Part 4: "Short Messaging";
Part 5: "Multimedia Messaging";
Part 6: "Payment";
Part 7: "Account management";
Part 8: "Terminal Status";
Part 9: "Terminal location";
Part 10: "Call handling";
Part 11: "Audio call";
Part 12: "Multimedia conference";
Part 13: "Address list management";
Part 14: "Presence".

3GPP
1 Scope

The present document is Part 4 of the Stage 3 Parlay X Web Services specification for Open Service Access (OSA).

The OSA specifications define an architecture that enables application developers to make use of network functionality through an open standardized interface, i.e. the OSA APIs. The concepts and the functional architecture for the OSA are contained in 3GPP TS 23.198 [3]. The requirements for OSA are contained in 3GPP TS 22.127 [2].

The present document specifies the Short Messaging Web Service aspects of the interface. All aspects of the Short Messaging Web Service are defined here, these being:

- Name spaces.
- Sequence diagrams.
- Data definitions.
- Interface specification plus detailed method descriptions.
- Fault definitions.
- Service policies.
- WSDL description of the interfaces.

The present document has been defined jointly between 3GPP TSG CN WG5, ETSI TISPAN and The Parlay Group.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document in the same Release as the present document.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".
[2] 3GPP TS 22.127: "Service Requirement for the Open Services Access (OSA); Stage 1".
[3] 3GPP TS 23.198: "Open Service Access (OSA); Stage 2".
[4] 3GPP TS 22.101: "Service aspects; Service principles".

NOTE: Available at http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/.

[6] 3GPP TS 29.199-1: "Open Service Access (OSA); Parlay X Web Services; Part 1: Common".
[7] 3GPP TS 23.040: "Technical realization of Short Message Service (SMS)".
3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in 3GPP TS 29.199-1 [6] apply. Additionally the following definition is needed:

**Shortcode**: a short telephone number, usually 4 to 6 digits long. This is represented by the ‘tel:’ URI defined in [6].

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TS 29.199-1 [6] and the following apply:

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMS</td>
<td>Short Message Service</td>
</tr>
<tr>
<td>SMS-C</td>
<td>Short Message Service - Center</td>
</tr>
</tbody>
</table>

4 Detailed service description

Currently, in order to programmatically receive and send SMS it is necessary to write applications using specific protocols to access SMS functions provided by network elements (e.g. SMS-C). This approach requires a high degree of network expertise. Alternatively it is possible to use the Parlay/OSA approach, invoking standard interfaces (e.g. User Interaction or Messaging Service Interfaces) to gain access to SMS capabilities, but these interfaces are usually perceived to be quite complex by IT application developers. Developers must have advanced telecommunication skills to use OSA interfaces.

In this clause is described a Parlay X Web Service, for sending and receiving SMS messages. The overall scope of this Web Service is to provide to application developers primitives to handle SMS in a simple way. In fact, using the SMS Web Service, application developers can invoke SMS functions without specific Telco knowledge.

ShortMessaging provides operations (see clause 8.1, Send SMS API) for sending a SMS message to the network and a polling mechanism for monitoring the delivery status of a sent SMS message. It is expected that a future release of this specification will also provide an asynchronous notification mechanism for delivery status.

ShortMessaging also allows an application to receive SMS messages. Both a polling (see clause 8.3, Receive SMS API) and an asynchronous notification mechanism (see clause 8.2, SMS Notification API) are available.

Figure 1 shows a scenario using the SMS Web Service to send an SMS message from an application. The application invokes a Web Service to retrieve a weather forecast for a subscriber (1) and (2) and a Parlay X Interface (3) to use the SMS Web Service operations (i.e. to send an SMS). After invocation, the SMS Web Service invokes a Parlay API method (4) using the Parlay/OSA SCS (Generic User Interaction) interface. This SCS handles the invocation and sends an UCP operation (5) to an SMS-C. Subsequently the weather forecast is delivered (6) to the subscriber.

In an alternative scenario, the Parlay API interaction involving steps (4) and (5) could be replaced with a direct interaction between the SMS Web Service and the Mobile network.
Figure 1: Send SMS Scenario

Figure 2 shows a scenario using the SMS Web Service to deliver a received SMS message to an application. The application receives a Parlay X Web Service invocation for an SMS sent by a subscriber (1) and (2). The SMS message contains the e-mail address of the person the user wishes to call. The application invokes a Parlay X Interface (3) to the Third Party Call Web Service in order to initiate the call (4).

Figure 2: Receive SMS Scenario
5 Namespaces

The SendSms interface uses the namespace:
www.csapi.org/wsd/parlayx/sms/send/v2_0

The ReceiveSms interface uses the namespace:
www.csapi.org/wsd/parlayx/sms/receive/v2_0

The SmsNotification interface uses the namespace:
www.csapi.org/wsd/parlayx/sms/notification/v2_0

The data types are defined in the namespace:
www.csapi.org/schema/parlayx/sms/v2_0

The 'xsd' namespace is used in the present document to refer to the XML Schema data types defined in XML Schema [5]. The use of the name 'xsd' is not semantically significant.

6 Sequence Diagrams

6.1 Send SMS and report status

Sending SMS message from Web portals is a common capability offered by Service Providers. This sequence diagram shows a portal providing this service.
7 XML Schema data type definition

7.1 DeliveryStatus enumeration

List of delivery status values.

<table>
<thead>
<tr>
<th>Enumeration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeliveredToNetwork</td>
<td>Successful delivery to network</td>
</tr>
<tr>
<td>DeliveryUncertain</td>
<td>Delivery status unknown: e.g. because it was handed off to another network.</td>
</tr>
<tr>
<td>DeliveryImpossible</td>
<td>Unsuccessful delivery; the message could not be delivered before it expired.</td>
</tr>
<tr>
<td>MessageWaiting</td>
<td>The message is still queued for delivery. This is a temporary state, pending transition to one of the preceding states.</td>
</tr>
<tr>
<td>DeliveredToTerminal</td>
<td>Successful delivered to Terminal</td>
</tr>
<tr>
<td>DeliveryNotificationNotSupported</td>
<td>Unable to provide delivery receipt notification. NotifySMDeliveryReceipt function will provide “DeliveryNotificationNotSupported” to indicate that delivery receipt for the specified address in a SendSMSRequest is not supported.</td>
</tr>
</tbody>
</table>

7.2 SmsFormat enumeration

List of SMS format values.
7.3 DeliveryInformation structure

Delivery status information.

<table>
<thead>
<tr>
<th>Element name</th>
<th>Element type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>xsd:anyURI</td>
<td>It indicates the destination address to which the notification is related.</td>
</tr>
<tr>
<td>DeliveryStatus</td>
<td>DeliveryStatus</td>
<td>Indicates the delivery result for destinationAddress. Possible values are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 'Delivered';</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 'DeliveryUncertain';</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 'DeliveryImpossible'.</td>
</tr>
</tbody>
</table>

7.4 SmsMessage structure

SMS message information. The SenderAddress is the address from which the message was actually sent, which may or may not match the senderName value provided in the SendSms operation.

<table>
<thead>
<tr>
<th>Element name</th>
<th>Element type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message</td>
<td>xsd:string</td>
<td>Text received in SMS</td>
</tr>
<tr>
<td>SenderAddress</td>
<td>xsd:anyURI</td>
<td>It indicates address sending the SMS</td>
</tr>
<tr>
<td>SmsServiceActivation</td>
<td>xsd:anyURI</td>
<td>Number associated with the invoked Message service, i.e. the destination address used to send the message</td>
</tr>
</tbody>
</table>

8 Web Service interface definition

8.1 Interface: SendSms

8.1.1 Operation: SendSms

The invocation of sendSms requests to send an SMS, specified by the String Message to the specified address (or address set), specified by Addresses. Optionally the application can also indicate the sender name (SenderName), i.e. the string that is displayed on the user's terminal as the originator of the message, the charging information and a ReceiptRequest. The ReceiptRequest which is a SimpleReference structure indicates the application endpoint, interface used for notification of delivery receipt and a correlator that uniquely identifies the sending request. By invoking this operation with the optional ReceiptRequest parameter the application requires to receive the notification of the status of the SMS delivery.

If Notification mechanism is not supported by a network a ServiceException (SVC0283) will be returned to the application and the message will not be sent to the addresses specified. Notification to the application is done by invoking the notifySMSDeliveryReceipt operation at the endpoint specified in ReceiptRequest.

The application can also explicitly invoke the getSmsDeliveryStatus using the RequestIdentifier returned by the sendSms invocation to get the delivery status.

Addresses may include group URIs as defined in the Address List Management specification. If groups are not supported, a PolicyException (POL0006) will be returned to the application.

For GSM systems, if Message contains characters not in the GSM 7-bit character set, the SMS is sent as a Unicode SMS.
If Message is longer than the maximum supported length (e.g. for GSM, 160 GSM 7-bit characters or 70 Unicode characters), the message will be sent as several concatenated short messages.

The correlator provided in the ReceiptRequest must be unique for this Web Service and application at the time the notification is initiated, otherwise a ServiceException (SVC0005) will be returned to the application.

8.1.1.1 Input message: SendSmsRequest

<table>
<thead>
<tr>
<th>Part name</th>
<th>Part type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addresses</td>
<td>xsd:anyURI [0..unbounded]</td>
<td>Addresses to which the SMS will be sent</td>
</tr>
<tr>
<td>SenderName</td>
<td>xsd:string</td>
<td>If present, it indicates the SMS sender name, i.e. the string that is</td>
</tr>
<tr>
<td></td>
<td></td>
<td>displayed on the user's terminal as the originator of the message</td>
</tr>
<tr>
<td>Charging</td>
<td>common:ChargingInformation</td>
<td>Charge to apply to this message (optional)</td>
</tr>
<tr>
<td>Message</td>
<td>xsd:string</td>
<td>Text to be sent in SMS</td>
</tr>
<tr>
<td>ReceiptRequest</td>
<td>common:SimpleReference</td>
<td>It defines the application endpoint, interfaceName and correlator that</td>
</tr>
<tr>
<td></td>
<td></td>
<td>will be used to notify the application when the message has been</td>
</tr>
<tr>
<td></td>
<td></td>
<td>delivered to terminal or if delivery is impossible (optional).</td>
</tr>
</tbody>
</table>

8.1.1.2 Output message : SendSmsResponse

<table>
<thead>
<tr>
<th>Part name</th>
<th>Part type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RequestIdentifier</td>
<td>xsd:string</td>
<td>it identifies a specific SMS delivery request</td>
</tr>
</tbody>
</table>

8.1.1.3 Referenced faults

ServiceException from [6]:
- SVC0001 - Service error.
- SVC0002 - Invalid input value.
- SVC0004 - No valid addresses.
- SVC0006 - Invalid group.
- SVC0280 - Message too long.
- SVC0283 – Delivery Receipt Notification not supported

PolicyException from [6]:
- POL0001 - Policy error.
- POL0006 - Groups not allowed.
- POL0007 - Nested groups not allowed.
- POL0008 - Charging not allowed.

8.1.2 Operation: SendSmsLogo

The invocation of \texttt{sendSmsLogo} requests to send an SMS logo, specified by the byte array \texttt{image} to the specified address (or address set), specified by \texttt{destinationAddressSet}. Optionally the application can also indicate the sender name (\texttt{senderName}), i.e. the string that is displayed on the user's terminal as the originator of the message, the charging information (\texttt{charging}) and a ReceiptRequest. The \texttt{receiptRequest} which is a SimpleReference structure indicates the application endpoint, interface used for notification of delivery receipt and a correlator that uniquely identifies the sending request. By invoking this operation with the optional \texttt{receiptRequest} parameter the application requires to receive the notification of the status of the SMS delivery.
If Notification mechanism is not supported by a network a serviceexception(SVC0283) will be returned to the application and the message will not be sent to the addresses specified. Notification to the application is done by invoking the notifySMSDeliveryReceipt operation at the endpoint specified in ReceiptRequest.

The application can also explicitly invoke the getSmsDeliveryStatus using the requestIdentifier returned by the sendSmsLogo invocation to get the delivery status.

Addresses may include group URIs as defined in the Address List Management specification. If groups are not supported, a PolicyException (POL0006) will be returned to the application.

The correlator provided in the ReceiptRequest must be unique for this Web Service and application at the time the notification is initiated, otherwise a ServiceException (SVC0005) will be returned to the application.

### 8.1.2.1 Input message: SendSmsLogoRequest

<table>
<thead>
<tr>
<th>Part name</th>
<th>Part type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addresses</td>
<td>xsd:anyURI [0..unbounded]</td>
<td>Addresses to which the SMS logo will be sent</td>
</tr>
<tr>
<td>SenderName</td>
<td>xsd:string</td>
<td>SMS sender name, i.e. the string that is displayed on the user’s terminal as the originator of the message (optional)</td>
</tr>
<tr>
<td>Charging</td>
<td>common:ChargingInformation</td>
<td>Charge to apply to this message (optional)</td>
</tr>
<tr>
<td>Image</td>
<td>xsd:base64Binary</td>
<td>The image in jpeg, gif or png format. The image will be scaled to the proper format</td>
</tr>
<tr>
<td>SmsFormat</td>
<td>SmsFormat</td>
<td>Possible values are: ‘Ems’ or ‘SmartMessaging’</td>
</tr>
<tr>
<td>ReceiptRequest</td>
<td>common:SimpleReference</td>
<td>It defines the application endpoint, interfaceName and correlator that will be used to notify the application when the message has been delivered to terminal or if delivery is impossible</td>
</tr>
</tbody>
</table>

### 8.1.2.2 Output message: SendSmsLogoResponse

<table>
<thead>
<tr>
<th>Part name</th>
<th>Part type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>requestIdentifier</td>
<td>String</td>
<td>It identifies a specific SMS delivery request</td>
</tr>
</tbody>
</table>

### 8.1.2.3 Referenced faults

ServiceException from [6]:
- SVC0001 - Service error.
- SVC0002 - Invalid input value.
- SVC0004 - No valid addresses.
- SVC0006 - Invalid group.
- SVC0281 - Unrecognized data format.
- SVC0283 – Delivery Receipt Notification not supported

PolicyException from [6]:
- POL.0001 - Policy error.
- POL.0006 - Groups not allowed.
- POL.0007 - Nested groups not allowed.
- POL.0008 - Charging not allowed.

### 8.1.3 Operation: SendSmsRingtone

The invocation of sendSmsRingtone requests to send an SMS ringtone, specified by the String ringtone (in RTX format) to the specified addresses, specified by Addresses. Optionally the application can also indicate the sender name.
(senderName) i.e. the string that is displayed on the user's terminal as the originator of the message, the charging
information (charging) and a receiptRequest. The receiptRequest which is a SimpleReference structure indicates the
application endpoint, interface used for notification of delivery receipt and a correlator that uniquely identifies the
sending request. By invoking this operation with the optional receiptRequest parameter the application requires to
receive the notification of the status of the SMS delivery.

If Notification mechanism is not supported by a network a serviceexception(SVC0283) will be returned to the
application and the message will not be sent to the addresses specified. Notification to the application is done by
invoking the notifySMSDeliveryReceipt operation at the endpoint specified in ReceiptRequest.

The application can also explicitly invoke the getSmsDeliveryStatus using the requestIdentifier returned by the
sendSMSRingTone invocation to get delivery status..

Addresses may include group URIs as defined in the Address List Management specification. If groups are not
supported, a PolicyException (POL0006) will be returned to the application.

The correlator provided in the ReceiptRequest must be unique for this Web Service and application at the time the
notification is initiated, otherwise a ServiceException (SVC0005) will be returned to the application.

Depending on the length of the ringtone, it may be sent as several concatenated short messages.

NOTE: On the RTX Ringtone Specification : An RTX file is a text file, containing the ringtone name, a control
subclause and a subclause containing a comma separated sequence of ring tone commands.

8.1.3.1 Input message: SendSmsRingtoneRequest

<table>
<thead>
<tr>
<th>Part name</th>
<th>Part type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addresses</td>
<td>xsd:anyURI [0..unbounded]</td>
<td>Addresses to which the SMS logo will be sent</td>
</tr>
<tr>
<td>SenderName</td>
<td>xsd:string</td>
<td>SMS sender name, i.e. the string that is displayed on the user’s terminal as the originator of the message (optional)</td>
</tr>
<tr>
<td>Charging</td>
<td>common:ChargingInformation</td>
<td>Charge to apply to this message (optional)</td>
</tr>
<tr>
<td>Ringtone</td>
<td>xsd:string</td>
<td>The ringtone in RTX format (see note above). (<a href="http://www.logomanager.co.uk/help/Edit/RTX.html">http://www.logomanager.co.uk/help/Edit/RTX.html</a>)</td>
</tr>
<tr>
<td>SmsFormat</td>
<td>SmsFormat</td>
<td>Possible values are: 'Ems' or 'SmartMessaging'</td>
</tr>
<tr>
<td>ReceiptRequest</td>
<td>common:SimpleReference</td>
<td>It defines the application endpoint, interfaceName and correlator that will be used to notify the application when the message has been delivered to terminal or if delivery is impossible</td>
</tr>
</tbody>
</table>

8.1.3.2 Output message: SendSmsRingtoneResponse

<table>
<thead>
<tr>
<th>Part name</th>
<th>Part type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RequestIdentifier</td>
<td>xsd:string</td>
<td>It identifies a specific SMS delivery request</td>
</tr>
</tbody>
</table>

8.1.3.3 Referenced faults

ServiceException from [6]:

- SVC0001 - Service error.
- SVC0002 - Invalid input value.
- SVC0004 - No valid addresses.
- SVC0006 - Invalid group.
- SVC0281 - Unrecognized data format.
- SVC0283 – Delivery Receipt Notification not supported

PolicyException from [6]:

- POL0001 - Policy error.
• POL0006 - Groups not allowed.
• POL0007 - Nested groups not allowed.
• POL0008 - Charging not allowed.

8.1.4 Operation: GetSmsDeliveryStatus

The invocation of getSmsDeliveryStatus requests the status of a previous SMS delivery request identified by requestIdentifier. The information on the status is returned in deliveryStatus, which is an array of status related to the request identified by requestIdentifier. The status is identified by a couplet indicating a user address and the associated delivery status. This method can be invoked multiple times by the application even if the status has reached a final value. However, after the status has reached a final value, status information will be available only for a limited period of time that should be specified in an off-line configuration step. The following four different SMS delivery status have been identified:

• 'DeliveredToNetwork': in case of concatenated messages, only when all the SMS-parts have been successfully delivered to the network.
• 'DeliveryUncertain': e.g. because it was handed off to another network.
• 'DeliveryImpossible': unsuccessful delivery; the message could not be delivered before it expired.
• 'MessageWaiting': the message is still queued for delivery.
• 'DeliveredToTerminal': in case of concatenated messages, only when all the SMS-parts have been successfully delivered to the terminal.

8.1.4.1 Input message: GetSmsDeliveryStatusRequest

<table>
<thead>
<tr>
<th>Part name</th>
<th>Part type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RequestIdentifier</td>
<td>xsd:string</td>
<td>It identifies a specific SMS delivery request</td>
</tr>
</tbody>
</table>

8.1.4.2 Output message: GetSmsDeliveryStatusResponse

<table>
<thead>
<tr>
<th>Part name</th>
<th>Part type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeliveryStatus</td>
<td>DeliveryInformation[0..unbounded]</td>
<td>It lists the variations on the delivery status of the SMS</td>
</tr>
</tbody>
</table>

8.1.4.3 Referenced faults

ServiceException from [6]:
• SVC0001 - Service error.
• SVC0002 - Invalid input value.

PolicyException from [6]:
• POL0001 - Policy error.

8.2 Interface: SmsNotification

SmsNotification is the application side notification interface to which short messages are delivered.

8.2.1 Operation: NotifySmsReception

The notification is used to send a short message to the application. The notification will occur only if the SMS fulfils the criteria specified when starting the SMS notification (see 8.4.1 Operation: StartSmsNotification).
The **notifySmsReception** method must be implemented by a Web Service at the application side. It will be invoked by the Parlay X server to notify the application of the reception of an SMS. The notification will occur if and only if the SMS received fulfils the criteria specified in an off-line provisioning step, identified by the **registrationIdentifier**. The criteria must at least include an **smsServiceActivationNumber**, i.e. the SMS destination address that can be “monitored” by the application. The parameter **senderAddress** contains the address of the sender. The application can apply the appropriate service logic to process the SMS.

### 8.2.1.1 Input message: NotifySmsReceptionRequest

<table>
<thead>
<tr>
<th>Part name</th>
<th>Part type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>correlator</td>
<td>xsd:string</td>
<td>Correlator provided in request to set up this notification</td>
</tr>
<tr>
<td>Message</td>
<td>SmsMessage</td>
<td>Message received</td>
</tr>
</tbody>
</table>

### 8.2.2 Operation: NotifySmsDeliveryReceipt

The **notifySmsDeliveryReceipt** method must be implemented by a Web Service at the application side if it requires notification of SMS delivery receipt. It will be invoked by the Parlay X server to notify the application when a SMS sent by an application has been delivered to the terminal of the recipient or if delivery is impossible. The notification will occur if and only if the status of the sent SMS is ‘DeliveredToTerminal’ or ‘DeliveryImpossible’ and the application has specified interest in notification when sending an SMS message by specifying the optional receiptRequest parameter. The correlator returned corresponds to the identifier specified by the application in the **receiptRequest** of the original **sendSMS** request.

When a SMS message is sent to multiple addresses, the notification from the server will send notification for each terminal as and when a SMS message is delivered to a terminal.

The following three different SMS delivery status will be returned in **NotifySMSDeliveryReceiptResponse**:

- **'DeliveryImpossible'**: unsuccessful delivery; the message could not be delivered before it expired.
- **'DeliveredToTerminal'**: in case of concatenated messages, only when all the SMS-parts have been successfully delivered to the terminal.
- **'DeliveredNotificationNotSupported'** - If notification is supported by the network but it does not support delivery receipt for one or more addresses specified in the **sendSMS** message. The service will send this status for those addresses.

### 8.2.2.1 Input message: NotifySmsDeliveryReceiptRequest

<table>
<thead>
<tr>
<th>Part name</th>
<th>Part type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correlator</td>
<td>xsd:string</td>
<td>The identifier defining the original SendRequest. This correlator was passed by the application during the SendSMS request</td>
</tr>
<tr>
<td>DeliveryStatus</td>
<td>DeliveryInformation</td>
<td>It lists the variations on the delivery status of the SMS to a terminal</td>
</tr>
</tbody>
</table>

### 8.2.2.2 Output message: NotifySmsDeliveryReceiptResponse

<table>
<thead>
<tr>
<th>Part name</th>
<th>Part type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>
8.2.2.3 Referenced faults

None.

8.3 Interface: ReceiveSms

8.3.1 Operation: GetReceivedSms

The invocation of getReceivedSms retrieves all the SMS messages received that fulfil the criteria identified by registrationIdentifier. The method returns only the list of SMS messages received since the previous invocation of the same method, i.e. each time the method is executed the messages returned are removed from the server. Moreover, each SMS message will be automatically removed from the server after a maximum time interval specified in an off-line configuration step.

The received SMS messages are returned in receivedSms. An SMS message is identified by a structure indicating the sender of the SMS message and the content.

8.3.1.1 Input message: GetReceivedSmsRequest

<table>
<thead>
<tr>
<th>Part name</th>
<th>Part type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RegistrationIdentifier</td>
<td>xsd:string</td>
<td>Identifies the off-line provisioning step that enables the application to receive notification of SMS reception according to specified criteria</td>
</tr>
</tbody>
</table>

8.3.1.2 Output message : GetReceivedSmsResponse

<table>
<thead>
<tr>
<th>Part name</th>
<th>Part type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReceivedSms</td>
<td>SmsMessage[0..unbounded]</td>
<td>It lists the received SMS since last invocation</td>
</tr>
</tbody>
</table>

8.3.1.3 Referenced faults

ServiceException from [6]:

- SVC0001 - Service error.
- SVC0002 - Invalid input value.

PolicyException from [6]:

- POL0001 - Policy error.

8.4 Interface: SmsNotificationManager

The short message notification manager enables applications to set up and tear down notifications for short messages.

8.4.1 Operation: StartSmsNotification

Start notifications to the application for a given SMS Service activation number and criteria.

The SMS Service activation number is an Address Data item as defined in 3GPP TS 29.199-1 [6]. A Shortcode is an example of an Address Data item.

The correlator provided in the reference must be unique for the application Web Service at the time the notification is initiated, otherwise a ServiceException (SVC0005) will be returned to the application.

If specified, criteria will be used to filter messages that are to be delivered to an application. If criteria are not provided, or is an empty string, then all messages for the SmsServiceActivationNumber will be delivered to the application.

The SmsServiceActivationNumber and criteria combination must be unique. If a criteria or the beginning parts of a
criteria overlaps then a fault will be returned to the application and the notification will not be set up. Note that the use of criteria will allow different notification endpoints to receive notifications for the same SmsServiceActivationNumber. The combination of SmsServiceActivationNumber and criteria must be unique, so that a notification will be delivered to only one notification endpoint. If no match is found, the message will not be delivered to the application.

8.4.1.1 Input message: StartSmsNotificationRequest

<table>
<thead>
<tr>
<th>Part name</th>
<th>Part type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>xsd:anyURI</td>
<td>the destination address to the short message</td>
</tr>
<tr>
<td>SmsServiceActivationNumber</td>
<td>xsd:string</td>
<td>Optional. The text to match against to determine the application to receive the notification. This text is matched against the first word of the short message body text</td>
</tr>
</tbody>
</table>

8.4.1.2 Output message: StartSmsNotificationResponse

<table>
<thead>
<tr>
<th>Part Name</th>
<th>Part Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

8.4.1.3 Referenced Faults

ServiceException from [6]

- SVC0001 – Service error
- SVC0002 – Invalid input value
- SVC0005 – Duplicate correlator
- SVC0282 – Overlapping Criteria

PolicyException from [6]

- POL0001 – Policy error

8.4.2 Operation: StopSmsNotification

The application may end a short message notification using this operation

8.4.2.1 Input message: StopSmsNotificationRequest

<table>
<thead>
<tr>
<th>Part name</th>
<th>Part type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reference</td>
<td>common:SimpleReference</td>
<td>Notification endpoint provided in request to set up the short message notification</td>
</tr>
</tbody>
</table>

8.4.2.2 Output message: StopSmsNotificationResponse

<table>
<thead>
<tr>
<th>Part Name</th>
<th>Part Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

8.4.2.3 Referenced Faults

ServiceException from [6]
• SVC0001 – Service error
• SVC0002 – Invalid input value
PolicyException from [6]
• POL0001 – Policy error

9 Fault definitions

9.1 ServiceException

9.1.1 SVC0280: Message too long

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Id</td>
<td>SVC0280</td>
</tr>
<tr>
<td>Text</td>
<td>Message too long. Maximum length is %1 characters</td>
</tr>
<tr>
<td>Variables</td>
<td>%1 Number of characters allowed in a message</td>
</tr>
</tbody>
</table>

9.1.2 SVC0281: Unrecognized data format

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Id</td>
<td>SVC0281</td>
</tr>
<tr>
<td>Text</td>
<td>Data format not recognized for message part %1</td>
</tr>
<tr>
<td>Variables</td>
<td>%1 Message part with the unrecognized data</td>
</tr>
</tbody>
</table>

9.1.3 SVC0282: Overlapping Criteria

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Id</td>
<td>SVC0282</td>
</tr>
<tr>
<td>Text</td>
<td>Overlapped Criteria %1</td>
</tr>
<tr>
<td>Variables</td>
<td>%1 Message part with the overlapped criteria</td>
</tr>
</tbody>
</table>

9.1.4 SVC0283: Delivery Receipt Notification not supported

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Id</td>
<td>SVC0283</td>
</tr>
<tr>
<td>Text</td>
<td>Delivery Receipt Notification not supported</td>
</tr>
<tr>
<td>Variables</td>
<td></td>
</tr>
</tbody>
</table>

10 Service policies

Service policies for this service.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GroupSupport</td>
<td>xsd:boolean</td>
<td>Groups may be included with addresses</td>
</tr>
<tr>
<td>NestedGroupSupport</td>
<td>xsd:boolean</td>
<td>Are nested groups supported in group definitions</td>
</tr>
<tr>
<td>ChargingSupported</td>
<td>xsd:boolean</td>
<td>Is charging supported for send operations</td>
</tr>
</tbody>
</table>
Annex A (normative):
WSDL for Short Messaging

The document/literal WSDL representation of this interface specification is compliant to 3GPP TS 29.199-1 [6] and is contained in text files (contained in archive 29199-04-610-doclit.zip) which accompanies the present document.
Annex B (informative):
Change history

<table>
<thead>
<tr>
<th>Date</th>
<th>TSG #</th>
<th>TSG Doc.</th>
<th>CR</th>
<th>Rev</th>
<th>Subject/Comment</th>
<th>Old</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dec 2003</td>
<td>CN_21</td>
<td>NP-030552</td>
<td>--</td>
<td>--</td>
<td>Submitted to CN#22 for Information</td>
<td>1.0.0</td>
<td></td>
</tr>
<tr>
<td>Jan 2004</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>Added The W3C WSDL representation of the APIs specified in the present document is contained in a set of files which accompany the present document: px0326rpcrenc.zip px0326rpclit.zip</td>
<td>1.0.1</td>
<td></td>
</tr>
<tr>
<td>Jun 2004</td>
<td>CN_24</td>
<td>NP-040274</td>
<td>--</td>
<td>--</td>
<td>Split into multi-part specification, 29.199-0n, for n=1,2,…9. Submitted to CN#24 for Information</td>
<td>1.0.3</td>
<td></td>
</tr>
<tr>
<td>Sep 2004</td>
<td>CN_25</td>
<td>NP-040380</td>
<td>--</td>
<td>--</td>
<td>Draft v2.0 submitted to TSG CN#25 for Approval.</td>
<td>2.0.0</td>
<td>6.0.0</td>
</tr>
<tr>
<td>Dec 2004</td>
<td>CN_26</td>
<td>NP-040407</td>
<td>001</td>
<td>--</td>
<td>Add SmsNotificationManager interface to PXWS Short-Messaging</td>
<td>6.0.0</td>
<td>6.1.0</td>
</tr>
</tbody>
</table>