3GPP TSG CN Plenary Meeting #26 **NP-040486**
08-10 December 2004, Athens, GREECE

Source: **CN5 (OSA)**

Title: **5 Rel-6 CR 29.199-xy OSA Parlay X Web Services**

Agenda item: **9.7 (OSA Enhancements [OSA3])**

Document for: **APPROVAL**

Doc-1st-Level	Spec	CR	Rev	Phase	Subject	Cat	Version-Current	Doc-2nd-Level	Workitem
NP-040487	29.199-04	001	--	Rel-6	Add SmsNotificationManager interface to PXWS Short-Messaging	B	6.0.0	N5-040878	OSA3
NP-040487	29.199-04	002	--	Rel-6	Add SmsNotificationManager interface to PXWS Short-Messaging	B	6.0.0	N5-040880	OSA3
NP-040487	29.199-05	001	--	Rel-6	Add MessageNotificationManager interface to PXWS Multimedia-Messaging	B	6.0.0	N5-040879	OSA3
NP-040487	29.199-09	001	--	Rel-6	Add PXWS Terminal Location Tracking Accuracy	C	6.0.0	N5-040784	OSA3
NP-040487	29.199-14	001	--	Rel-6	Correct the Presence WSDL source code	F	6.0.0	N5-040871	OSA3

CR-Form-v7

CHANGE REQUEST

⌘ 29.199-04 CR 001 ⌘rev - ⌘ Current version: 6.0.0 ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: | UICC apps⌘ ☐ ME ☐ Radio Access Network ☐ Core Network X

Title:	⌘	Add SmsNotificationManager interface to PXWS Short-Messaging
Source:	⌘	CN5 Incomit, Telenor
Work item code:	⌘	OSA3
Date:	⌘	30/11/2004
Category:	⌘	**B**
Release:	⌘	*REL-6*

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*

Reason for change:	⌘	On-line SmsNotification request is missing in the current specification
Summary of change:	⌘	Addition of the SmsNotificationManager interface
Consequences if not approved:	⌘	1- The Notifcation Interface can only be initiated off-line. 2- Problems with Application portability among deployment. 3- Propriatary notification management. 4- Inconsistency with other ParlayX API.
Clauses affected:	⌘	3,1, 4, 8.2, New 8.4, 9.1 , Annex A

Other specs affected: ⌘

Y	N		
	X	Other core specifications	⌘
	X	Test specifications	
	X	O&M Specifications	

Other comments: ⌘

Changes in Clause 3.1

3.1 Definitions

For the purposes of the present document, the terms and definitions given in 3GPP TS 29.199-1 [6] apply.

Additionlly 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 3GPP TS 29.199-1 [6]].

End Changes in Clause 3.1

Changes in Clause 4

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) ~~F~~for sending a SMS message to the network ~~(see clause 8.1 of the present document, Send SMS API~~and a polling mechanism for ~~), the application invokes a message to send it and~~monitoring the delivery status of a sent SMS message ~~must subsequently become active again to poll for delivery status.~~. ~~-~~ It is expected that a future release of this specification will also ~~There is an alternative to this polling mechanism, i.e. an asynchronous notification mechanism implemented with an application-side Web Service. However it was decided not to~~ provide ana ~~notification mechanism~~ asynchronous notification mechanism for delivery status ~~in the first release, to make the API as simple as possible, even though the polling mechanism is not as network efficient as the notification mechanism.~~.

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.

~~For receiving a message from the network, the application may use either polling (see clause 8.3 of the present document, Receive SMS API) or notification (see clause 8.2 of the present document, SMS Notification API) mechanisms. The notification mechanism is more common: network-initiated messages are sent to autonomous application-side Web Services. The Notification is requested via the SmsNotificationManager interface or via a specific off-line mechanism.~~

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 ~~SMS~~ (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.

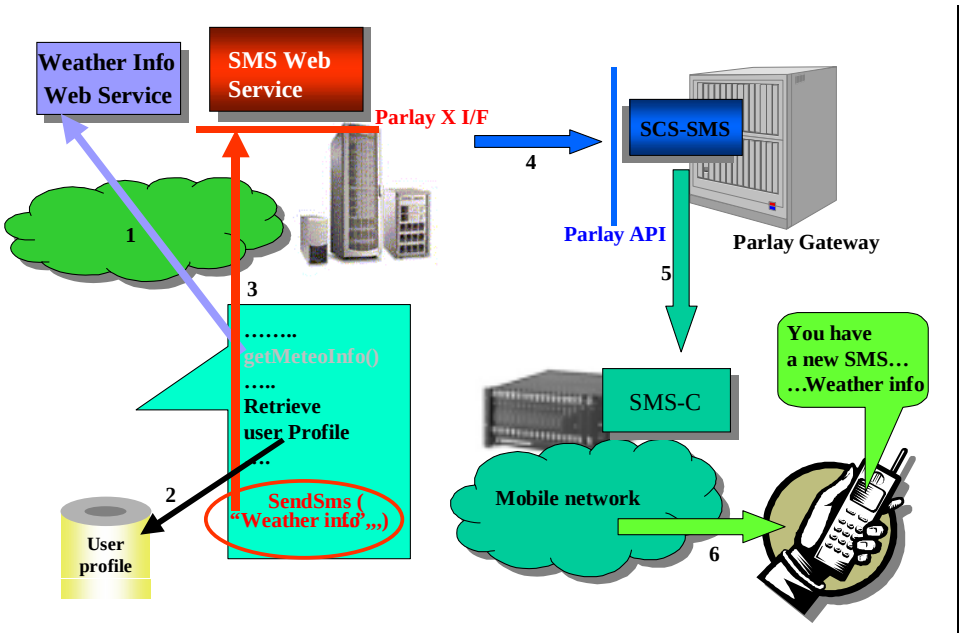

Figure 1: Send SMS Scenario

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 ~~to retrieve~~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).

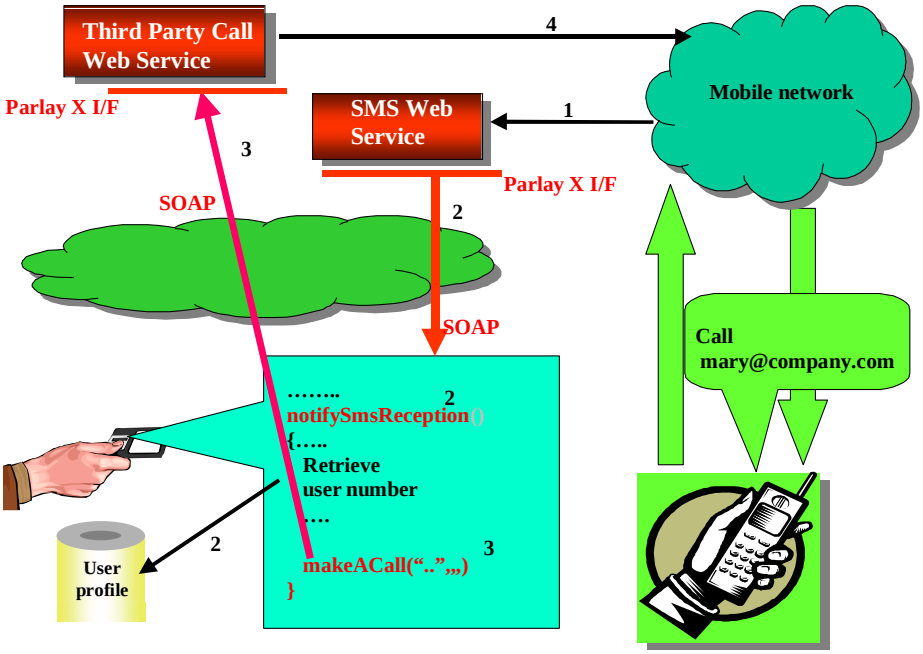

Figure 2: Receive SMS Scenario

End Changes in Clause 4

Changes in Clause 8.2

8.2 Interface: SmsNotification

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

8.2.1 Operation: NotifySmsReception

~~The **notifySmsReception** method must be implemented by a Web Service at the ***application side***. It will be invoked by the Parlay X server~~The notification is used to ~~to notify the application of the reception of an~~send a short message to the application ~~SMS.~~. The notification will occur ~~if and~~ only if the SMS ~~received~~ fulfils the criteria specified when starting the SMS notification (see ~~in an off-line provisioning step, identified by the **registrationIdentifier**~~8.4.1 Operation: StartSmsNotification)~~**.** 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

Part name	Part type	Description
~~RegistrationIdentifier~~correlator	xsd:string	~~Identifies the off-line on-line as described in 8.4 provisioning step that enables the application to receive notification of SMS reception according to specified criteria~~ Correlator provided in request to set up this notification
Message	SmsMessage	Message received

End Changes in Clause 8.2

Clause 8.4

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 DES-TISPAN-01007-01 [3]. 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

Part name	Part type	Description
Reference	common:SimpleReference	Notification endpoint definition
SmsServiceActivation Number	xsd:anyURI	the destination address to the short message
Criteria	xsd:string	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

8.4.1.2 Output message: StartSmsNotificationResponse

Part Name	Part Type	Description
none		

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

Part name	Part type	Description
reference	common:SimpleReference	Notification endpoint provided in request to set up the short message notification

8.4.2.2 Output message: StopSmsNotificationResponse

Part Name	Part Type	Description
None		

8.4.2.3 Referenced Faults

ServiceException from [6]

- SVC0001 – Service error

- SVC0002 – Invalid input value

PolicyException from [6]

- POL0001 – Policy error

End Clause 8.4

Clause 9.1

9 Fault definitions

9.1 ServiceException

9.1.1 SVC0280: Message too long

Name	Description
Message Id	SVC0280
Text	Message too long. Maximum length is %1 characters
Variables	%1 Number of characters allowed in a message

9.1.2 SVC0281: Unrecognized data format

Name	Description
Message Id	SVC0281
Text	Data format not recognized for message part %1
Variables	%1 Message part with the unrecognized data

9.1.3 SVC0282: Overlapping Criteria

Name	Description
Message Id	SVC0282
Text	Overlapped Criteria %1
Variables	%1 Message part with the overlapped criteria

End Clause 9.1

Begin change in Annex A

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-~~600~~610-doclit.zip) which accompanies the present document.

End of Change in Annex A
End of Document

joint-API-group (Parlay, ETSI Project OSA, 3GPP TSG_CN WG5) **N5-040880**
Meeting #29, Barcelona, SPAIN, 01-05 November 2004

CR-Form-v7

CHANGE REQUEST

⌘ **29.199-04** **CR** **CRNum** ⌘**rev** **-** ⌘ Current version: **6.0.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: | UICC apps⌘ ☐ ME ☐ Radio Access Network ☐ Core Network **X**

Title:	⌘	Add PXWS SMS Notification Delivery Reception
Source:	⌘	CN5 Orange
Work item code:	⌘	OSA3
Date:	⌘	01/12/2004
Category:	⌘	**B**
Release:	⌘	*REL-6*

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*

Reason for change:	⌘	Applications should be able to receive notification when a SMS is delivered to the terminal.
Summary of change:	⌘	Addition of notification interface for delivery reception and a deliveryreceptionNotification to sendSMS operations
Consequences if not approved:	⌘	Lack of above interface functions limit application development

Clauses affected:	⌘	7.1, 8.1, New 8.2.2, New 9.1.3, Annex A

	Y	N		
Other specs affected:	⌘		**X**	Other core specifications ⌘
			X	Test specifications
			X	O&M Specifications

Other comments:	⌘	

Change in Clause 7.1

7.1 DeliveryStatus enumeration

List of delivery status values.

Enumeration	Description
DeliveredToNetwork	Successful delivery to network
DeliveryUncertain	Delivery status unknown: 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. This is a temporary state, pending transition to one of the preceding states.
DeliveredToTerminal	Successful delivered to Terminal
DeliveryNotificationNotSupported	Unable to provide delivery receipt notification. NotifySMSDeliveryReceipt function will provide "DeliveryNotificationNotSupported" to indicate that delivery receipt for the specified address in a SendSMSRequest is not supported.

End of Change in Clause 7.1

Change in Clause 8.1.1

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, ~~and~~ 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(SVC0282) 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 **notifiySMSDeliveryReceipt** operation at the endpoint specified in ReceiptRequest.

~~In order to receive this information the~~The application can also ~~has~~ ~~to~~ explicitly invoke the **getSmsDeliveryStatus** using t~~. T~~he **RequestIdentifier**~~,~~ returned by the **sendSMS** invocation~~, can be used to identify the SMS delivery request~~ 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

Part name	Part type	Description
Addresses	xsd:anyURI [0..unbounded]	Addresses to which the SMS will be sent
SenderName	xsd:string	If present, it indicates the SMS sender name, i.e. the string that is displayed on the user's terminal as the originator of the message
Charging	common:ChargingInformation	Charge to apply to this message (optional)
Message	xsd:string	Text to be sent in SMS
ReceiptRequest	common:SimpleReference	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(Optional).

8.1.1.2 Output message : SendSmsResponse

Part name	Part type	Description
RequestIdentifier	xsd:string	It identifies a specific SMS delivery request

8.1.1.3 Referenced faults

ServiceException from DES-TISPAN-01007-01 [3]:

- SVC0001 - Service error.
- SVC0002 - Invalid input value.
- SVC0004 - No valid addresses.
- SVC0006 - Invalid group.
- SVC0280 - Message too long.
- SVC0282 – Delivery Receipt Notification not supported

PolicyException from DES-TISPAN-01007-01 [3]:

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

End of Change in Clause 8.1.1

Change in Clause 8.1.2

8.1.2 Operation: SendSmsLogo

The invocation of **sendSmsLogo** requests to send an SMS logo, specified by the byte array **image** to the specified address (or address set), specified by **destinationAddressSet**. 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, ~~and~~ 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(SVC0282) 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 **notifiySMSDeliveryReceipt** operation at the endpoint specified in ReceiptRequest.

~~In order to receive this information t~~The application ~~has to~~can also explicitly invoke the **getSmsDeliveryStatus** using t~~.~~ ~~T~~he **requestIdentifier**~~,~~ returned by the **sendSMSLogo** invocation~~, can be used to identify the SMS delivery request~~ 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

Part name	Part type	Description
Addresses	xsd:anyURI [0..unbounded]	Addresses to which the SMS logo will be sent
SenderName	xsd:string	SMS sender name, i.e. the string that is displayed on the user's terminal as the originator of the message (optional)
Charging	common:ChargingInformation	Charge to apply to this message (optional)
Image	xsd:base64Binary	The image in jpeg, gif or png format. The image will be scaled to the proper format
SmsFormat	SmsFormat	Possible values are: 'Ems' or 'SmartMessaging'
ReceiptRequest	common:SimpleReference	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

8.1.2.2 Output message: SendSmsLogoResponse

Part name	Part type	Description
requestIdentifier	String	It identifies a specific SMS delivery request

8.1.2.3 Referenced faults

ServiceException from DES-TISPAN-01007-01 [3]:

- SVC0001 - Service error.
- SVC0002 - Invalid input value.
- SVC0004 - No valid addresses.
- SVC0006 - Invalid group.
- SVC0281 - Unrecognized data format.

- SVC0282 – Delivery Receipt Notification not supported

PolicyException from DES-TISPAN-01007-01 [3]:

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

End of Change in Clause 8.1.2

Change in Clause 8.1.3

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, ~~and~~ 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(SVC0282) 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 **notifiySMSDeliveryReceipt** operation at the endpoint specified in ReceiptRequest.

~~In order to receive this information t~~The application ~~has to~~can also explicitly invoke the **getSmsDeliveryStatus** using~~.~~ t~~T~~he **requestIdentifier**~~,~~ returned by the **sendSMSRingTone** invocation to get delivery status.~~, can be used to identify the SMS delivery request.~~ ~~.~~

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

Part name	Part type	Description
Addresses	xsd:anyURI [0..unbounded]	Addresses to which the SMS logo will be sent
SenderName	xsd:string	SMS sender name, i.e. the string that is displayed on the user's terminal as the originator of the message (optional)
Charging	common:ChargingInformation	Charge to apply to this message (optional)
Ringtone	xsd:string	The ringtone in RTX format (see note above). (http://www.logomanager.co.uk/help/Edit/RTX.html)
SmsFormat	SmsFormat	Possible values are: 'Ems' or 'SmartMessaging'
ReceiptRequest	common:SimpleReference	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

8.1.3.2 Output message: SendSmsRingtoneResponse

Part name	Part type	Description
RequestIdentifier	xsd:string	It identifies a specific SMS delivery request

8.1.3.3 Referenced faults

ServiceException from DES-TISPAN-01007-01 [3]:

- SVC0001 - Service error.
- SVC0002 - Invalid input value.
- SVC0004 - No valid addresses.
- SVC0006 - Invalid group.
- SVC0281 - Unrecognized data format.
- SVC0282 – Delivery Receipt Notification not supported

PolicyException from DES-TISPAN-01007-01 [3]:

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

End of Change in Clause 8.1.3

Change in Clause 8.1.4

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

End of Change in Clause 8.1

Begin of new Clause 8.2.2

8.2.2 Operation: NotifySmsDeliveryReceipt

The **notifySmsDeliveryReceipt** method must be implemented by a Web Service at the ***application side*** if it requires notification of SMSdelivery 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

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

8.2.2.2 Output message: NotifySmsDeliveryReceiptResponse

Part name	Part type	Description
None		

8.2.2.3 Referenced faults

None.

End of New Clause 8.2.2

Begin of new Clause 9.1.3

9.1.3 SVC0282: Delivery Receipt Notification not supported

Name	Description
Message Id	SVC0282
Text	Delivery Receipt Notification not supported
Variables	

End of New Clause 9.1.3
End of Document

Begin change in Annex A

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-~~600~~610-doclit.zip) which accompanies the present document.

End of Change in Annex A
End of Document

CR-Form-v7

CHANGE REQUEST

⌘ 29.199-05 CR CRNum ⌘rev - ⌘ Current version: 6.0.0 ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network ☐ Core Network X

Title:	⌘	Add MessageNotificationManager interface to PXWS Multimedia-Messaging
Source:	⌘	CN5 Incomit, Telenor
Work item code:	⌘	OSA3
Date:	⌘	30/11/2004
Category:	⌘	B
Release:	⌘	*REL-6*

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*

Reason for change:	⌘	Requesting the MessageNotification is missing in the current specification
Summary of change:	⌘	Addition of the MessageNotificationManager interface
Consequences if not approved:	⌘	1- The Notifcation Interface can only be initiated off-line. 2- Problems with Application portability among deployment. 3- Propriatary notification management. 4- Inconsistency with other ParlayX specification.
Clauses affected:	⌘	3.1, 4, 8.3, New 8.4, 9 , Annex A

	Y	N		
Other specs affected: ⌘		X	Other core specifications	⌘
		X	Test specifications	
		X	O&M Specifications	

Other comments: ⌘

Changes in Clause 3.1

3.1 Definitions

For the purposes of the present document, the terms and definitions given in 3GPP TS 29.199-1 [6] apply.

Additionlly the following definition is needed:

Shortcode: a short telephone number usually 4 to 6 digits long, this is represented by the the 'tel:' URI defined in 3GPP TS 29.199-1 [6].

End Changes in Clause 3.1

Changes in Clause 4

4 Detailed service description

Currently, in order to programmatically receive and send Multimedia Messages, it is necessary to write applications using specific protocols to access MMS functions provided by network elements (e.g. MMS-C). This approach requires application developers to have a high degree of network expertise.

This contribution defines a Multimedia Messaging Web Service that can map to SMS, EMS, MMS, IM, E-mail, etc.

The choice is between defining one set of interfaces per messaging network or a single set common to all networks; e.g. we could define sendMMS, sendEMS, sendSMS, etc., or just use sendMessage. Although the more specific the API the easier it is to use, there are advantages to a single set of network-neutral APIs. These advantages include:

- improved service portability;
- lower complexity, by providing support for generic user terminal capabilities only.

For this version of the Parlay X specification, we provide sets of interfaces for two messaging Web Services: Short Messaging (part 4) and Multimedia Messaging (this part), which provides generic messaging features (including SMS).

~~For sending a message to the network (see Send Message), the application invokes a message to send it and must subsequently become active again to poll for delivery status. There is an alternative to this polling mechanism, i.e. an asynchronous notification mechanism implemented with an application-side Web Service. However it was decided not to provide a notification mechanism in the first release, to make the interface as simple as possible, even though the polling mechanism is not as network efficient as the notification mechanism.~~

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

Multimedia Messaging also allows an application to receive Multimedia messages. Both a polling (see clause 8.2, ReceiveMessage API) and an asynchronous notification mechanism (see clause 8.3, Message Notification API) are available.

~~For receiving a message from the network, the application may use either polling (see Receive Message) or notification (see Message Notification) mechanisms. The notification mechanism is more common: network-initiated messages are sent to autonomous application-side Web Services. Both mechanisms are supported, but the provisioning of the notification-related criteria is not specified.~~

Figure **Error! Reference source not found.** shows an example scenario using sendMessage and getMessageDeliveryStatus to send data to subscribers and to determine if the data has been received by the subscriber. The application invokes a Web Service to retrieve a stock quote (1) and (2) and sends the current quote - sendMessage - using the Parlay X Interface (3) of the Multimedia Messaging Web Service. After invocation, the Multimedia Message Web Service sends the message to an MMS-C using the MM7 interface (4) for onward transmission (5) to the subscriber over the Mobile network.

Later, when the next quote is ready, the application checks to see - getMessageDeliveryStatus - if the previous quote has been successfully delivered to the subscriber. If not, it may for instance perform an action (not shown) to provide a credit for the previous message transmission. This way, the subscriber is only charged for a stock quote if it is delivered on time.

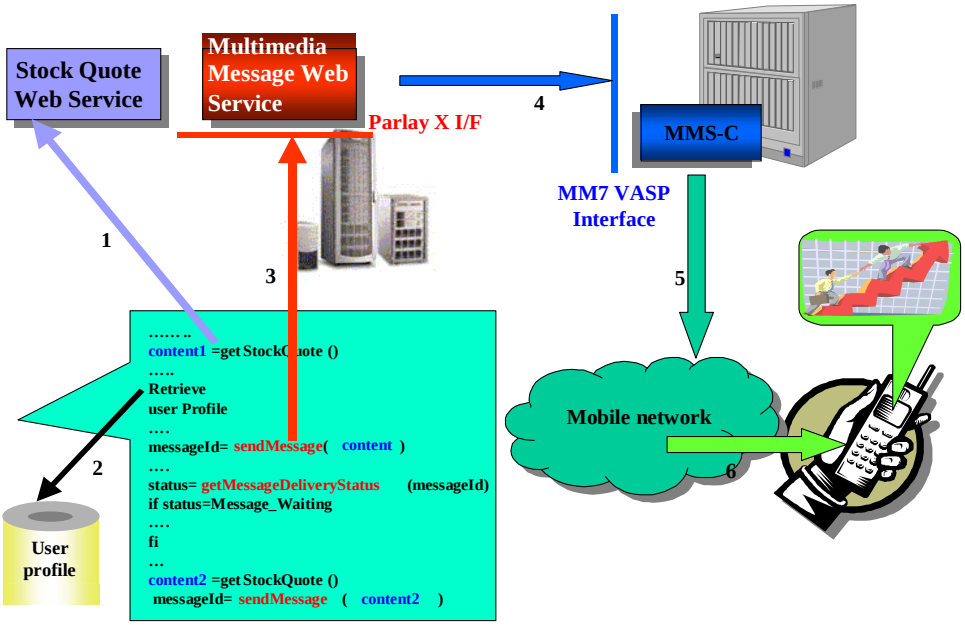

Figure 1: Multimedia Messaging Scenario

End Changes in Clause 4

Changes in Clause 8.3

8.3 Interface: MessageNotification

MessageNotification is the application side notification interface to which multimedia messages are delivered.

8.3.1 Operation: NotifyMessageReception

The notification is used to send a multimedia message to the application. The notification will occur only if the multimedia message fulfils the criteria specified when starting the multimedia message notification (See 8.4.1 Start MessageNotification).

~~This method will have to be implemented by a Web Service on the client application side. The registration of the URI for this application Web Service is done off-line. This means that there is a registration mechanism in the Parlay X Gateway that binds different **registrationIdentifier** parameters to applications and their Web Service URIs.~~

~~A client application is notified that a new Message, sent to a specific Service Activation Number, has been received. Using the **registrationIdentifier**, the client application can apply appropriate service logic with specific behaviour.~~

8.3.1.1 Input message: NotifyMessageReceptionRequest

Part name	Part type	Description
~~RegistrationIdentifier~~ correlator	xsd:string	Correlator provided in request to set up this notification ~~A handle connected to the off-line on-line registration of the notifications. This distinguishes registrations that point to the same application Web Service.~~
Message	MessageReference	This parameter contains all the information associated with the received message.

End Changes in Clause 8.3

New Clause 8.4

8.4 Interface: MessageNotificationManager

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

8.4.1 Operation: StartMessageNotification

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

The Message 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 MessageServiceActivationNumber will be delivered to the application. The MessageServiceActivationNumber 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 MessageServiceActivationNumber. The combination of MessageServiceActivationNumber 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: StartMessageNotificationRequest

Part name	Part type	Description
Reference	common:SimpleReference	Notification endpoint definition
MessageServiceActivationNumber	xsd:anyURI	the destination address of the multimedia message
Criteria	xsd:string	Optional. The text to match against to determine the application to receive the notification. This text is matched against the first word of the subject of the multimedia message or the first word in the text part of the multimedia message

8.4.1.2 Output message: StartMessageNotificationResponse

Part Name	Part Type	Description
none		

8.4.1.3 Referenced Faults

ServiceException from [6]

- SVC0001 – Service error
- SVC0002 – Invalid input value
- SVC0005 – Duplicate correlator
- SVC0230 – Overlapping Criteria

PolicyException from [6]

- POL0001 – Policy error

8.4.2 Operation: StopMessageNotification

The application may end a multimedia message notification using this operation

8.4.2.1 Input message: StopMessageNotificationRequest

Part name	Part type	Description
Reference:	common:SimpleReference	Notification endpoint provided in request to set up the multimedia message notification

8.4.2.2 Output message: StopMessageNotificationResponse

Part Name	Part Type	Description
None		

8.4.2.3 Referenced Faults

ServiceException from [6]

- SVC0001 – Service error
- SVC0002 – Invalid input value

PolicyException from [6]

- POL0001 – Policy error

End of New Clause 8.4

Changes in Clause 9

9 Fault definitions

~~No new faults are defined by this service.~~

9.1 ServiceException

9.1.1 SVC0230: Overlapping Criteria

Name	Description
Message Id	SVC0230
Text	Overlapped Criteria %1
Variables	%1 Message part with the overlapped criteria

End Changes in Clause 9

Begin change in Annex A

Annex A (normative): WSDL for Multimedia 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 29199-05-~~600~~610-doclit.zip) which accompanies the present document.

End of Change in Annex A
End of Document

CR-Form-v7

CHANGE REQUEST

⌘ **29.199-09** **CR** **001** ⌘ **rev** **-** ⌘ Current version: **6.0.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network ☐ Core Network **X**

Title:	⌘	Add PXWS Terminal Location Tracking Accuracy
Source:	⌘	CN5 IBM, Sprint, Telenor, Incomit
Work item code:	⌘	OSA3
Date:	⌘	19/10/2004
Category:	⌘	**C**
Release:	⌘	*REL-6*

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*

Reason for change:	⌘	Accuracy is missing from the notification setup, reducing the usefulness of this interface and making it inconsistent with other operations defined within this specification.
Summary of change:	⌘	Addition of tracking accuracy message part to notification setup
Consequences if not approved:	⌘	Lack of tracking accuracy lowers the usefulness of the location notification since defaults must be used for accuracy. Inconsistency with other operations within this specification makes this interface more difficult to understand and use.

Clauses affected:	⌘	7.2, 8.2.1, 10

		Y	N		
Other specs affected:	⌘		X	Other core specifications	⌘
			X	Test specifications	
			X	O&M Specifications	

Other comments:	⌘	

Change in Clause 7.2

7.2 Accuracy values

Two accuracy values are used in some of the operations. These values express the desire of the application for the location information to be provided by the Web Service. The choice of values may influence the price that the Service Provider charges.

The 'requested accuracy' expresses the range in which the application wishes to receive location information. This may influence the choice of location technology to use (for instance, cell sector location may be suitable for requests specifying 1 000 meters, but GPS technology may be required for requests below 100 meters).

The 'acceptable accuracy' expresses the range that the application considers useful - if the location cannot be determined within this range, then the application would prefer not to receive the information. For instance, a taxi tracking service to determine the closest taxi to a person may not be useful if the accuracy cannot be provided within 1 000 meters to provide prompt service. This will also reduce customer satisfaction issues, since results that are not useful can be handled appropriately for billing (e.g. Service Provider may choose not to bill for these).

In triggered notifications, a tracking accuracy is defined. This accuracy refers not to the accuracy for the area being checked against, but rather for the accuracy of the technology used to track the terminal. For instance, a fine grained tracking accuracy would be suitable for tracking the terminal entering a specific location, like a person arriving at a destination building. A coarse grained tracking accuracy would be appropriate for determining when a person has arrived at a city after a plane trip or a truck nearing the vicinity of a warehouse.

End of Change in Clause 7.2

Change in Clause 8.2.1

8.2.1 Operation: StartGeographicalNotification

Notifications of location changes are made available to applications. The number and duration of notifications may be requested as part of the setup of the notification or may be governed by service policies, or a combination of the two.

If CheckImmediate is set to true, then the notification will be set up, and then the current value of the terminal location will be checked. If the terminal location is within the radius provided and the criteria is Entering or is outside the radius and the criteria is Leaving, a notification will be sent to the application. This notification will count against the count requested. This addresses the case where the location of the device changes during the time the notification is being set up, which may be appropriate in some applications.

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

If the frequency requested is more often than allowed by the service policy, then the value in the service policy will be used. If the duration requested exceeds the time allowed in the service policy, then the value in the service policy will be used. If the notification period (duration) ends before all of the notifications (count) have been delivered, then the notification terminates. In all cases, when the notifications have run their course (by duration or count), an end of notifications message will be provided to the application.

Service policies may govern what count values can be requested, including maximum number of notifications allowed and whether unlimited notifications can be requested (specifying a count of zero). If the count value provided is not in policy, a PolicyException (POL0004 or POL0005 as appropriate) will be returned.

The criteria will be met when the terminal enters the area defined as the circle of the radius provided around the point provided (latitude, longitude). The tracking accuracy provided will determine how fine grained the determination of where the terminal is at is. A tracking accuracy with a high value (coarse grained tracking) may result in more or less notifications (false notifications or missed notifications) than actual entries and exits from the area defined.

Service policies govern what values can be provided for tracking accuracy, including a minimum number of meters for tracking accuracy that can be requested. If the value provided is not within policy, a PolicyException (POL0230) will be returned.

Input message: StartGeographicalNotificationRequest

Part name	Part type	Description
Reference	common:SimpleReference	Notification endpoint definition
Addresses	xsd:anyURI [0..unbounded]	Addresses of terminals to monitor
Latitude	xsd:float	Latitude of center point
Longitude	xsd:float	Longitude of center point
Radius	xsd:float	Radius of circle around center point in meters
TrackingAccuracy	xsd:float	Number of meters of acceptable error in tracking distance
Criteria	EnteringLeavingCriteria	Indicates whether the notification should occur when the terminal enters or leaves the target area
CheckImmediate	xsd:boolean	Check location immediately after establishing notification
Frequency	common:TimeMetric	Maximum frequency of notifications (can also be considered minimum time between notifications)
Duration	common:TimeMetric	Length of time notifications occur for, null to use default notification time defined by service policy
Count	xsd:int	Maximum number of notifications, zero if no maximum

8.2.1.2 Output message: StartGeographicalNotificationResponse

Part name	Part type	Description
None		

8.2.1.3 Referenced faults

ServiceException from 3GPP TS 29.199-1 [6]:

- SVC0001: Service error.
- SVC0002: Invalid input value.
- SVC0004: No valid addresses.
- SVC0005: Duplicate correlator.
- SVC0006: Invalid group.

PolicyException from 3GPP TS 29.199-1 [6]:

- POL0001: Policy error.
- POL0003: Too many addresses.
- POL0004: Unlimited notifications not supported.
- POL0005: Too many notifications requested.
- POL0006: Groups not allowed.
- POL0007: Nested groups not allowed.
- POL0009: Invalid frequency requested.
- POL0230: Requested accuracy not available
- POL0231: Geographic notification not available.

End of Change in Clause 8.2.1

Change in Clause Clause 10

Name	Type	Description
MinimumAccuracy	xsd:int	Minimum value for requested accuracy
MinimumAcceptableAccuracy	xsd:int	Minimum value for acceptable accuracy
MinimumTrackingAccuracy	xsd:int	Minimum value for tracking accuracy
GeographicalNotificationAvailable	xsd:boolean	Can notifications be set on a geography
PeriodicNotificationAvailable	xsd:boolean	Can a periodic notification be set up
AltitudeAlwaysAvailable	xsd:boolean	Is altitude available for all location responses
AltitudeSometimesAvailable	xsd:boolean	Is altitude available for some or all location responses (if AltitudeAlwaysAvailable is true, this is also true)
MaximumNotificationAddresses	xsd:int	Maximum number of addresses for which a notification can be set up
MaximumNotificationFrequency	common:TimeMetric	Maximum rate of notification delivery (also can be considered minimum time between notifications)
MaximumNotificationDuration	common:TimeMetric	Maximum amount of time a notification may be set up for
MaximumCount	xsd:int	Maximum number of notifications that may be requested
UnlimitedCountAllowed	xsd:boolean	Allowed to specify unlimited notification count (i.e. specify zero in notification count requested)
GroupSupport	xsd:boolean	Groups URIs may be used
NestedGroupSupport	xsd:boolean	Are nested groups supported in group definitions

End of Change in Clause 10
End of Document

CR-Form-v7

CHANGE REQUEST

⌘ **29.199-14** **CR** **001** ⌘ **rev** **-** ⌘ Current version: **6.0.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network ☐ Core Network **X**

Title:	⌘	Correct the Presence WSDL source code
Source:	⌘	CN5 (Joe McIntyre, IBM)
Work item code:	⌘	OSA3
Date:	⌘	18/11/2004
Category:	⌘	**F**
Release:	⌘	*REL-6*

Use one of the following categories:
- ***F*** *(correction)*
- ***A*** *(corresponds to a correction in an earlier release)*
- ***B*** *(addition of feature),*
- ***C*** *(functional modification of feature)*
- ***D*** *(editorial modification)*

Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*

Reason for change:	⌘	The Presence Parlay X Web Service WSDL source code was not syncrhonized with the Word TS 29.199-14
Summary of change:	⌘	The WSDL files packages in the archive was synchronized with the Word TS describing them.
Consequences if not approved:	⌘	Potential incompatible implementations of the Presence Parlay X Web Service

Clauses affected:	⌘	

Other specs affected: ⌘

Y	N	
	X	Other core specifications
	X	Test specifications
	X	O&M Specifications

⌘

Other comments: ⌘

Change in Annex A (normative):
WSDL of Presence API

Annex A (normative):
WSDL of Presence API

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 29199-14-6~~0~~1<u>1</u>0-doclit.zip) which accompany the present document.

End of Change in Annex A (normative):
WSDL of Presence API