

ETSI TS 129 199-5 V6.2.0 (2005-03)

Technical Specification

**Universal Mobile Telecommunications System (UMTS);
Open Service Access (OSA);
Parlay X web services;
Part 5: Multimedia messaging
(3GPP TS 29.199-05 version 6.2.0 Release 6)**



Reference

RTS/TSGC-0529199-05v620

Keywords

UMTS

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:

<http://www.etsi.org>

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at

<http://portal.etsi.org/tb/status/status.asp>

If you find errors in the present document, please send your comment to one of the following services:

http://portal.etsi.org/chaicor/ETSI_support.asp

Copyright Notification

No part may be reproduced except as authorized by written permission.
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2005.
All rights reserved.

DECTTM, **PLUGTESTS**TM and **UMTS**TM are Trade Marks of ETSI registered for the benefit of its Members.
TIPHONTM and the **TIPHON logo** are Trade Marks currently being registered by ETSI for the benefit of its Members.
3GPPTM is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (<http://webapp.etsi.org/IPR/home.asp>).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

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

The present document may refer to technical specifications or reports using their 3GPP identities, UMTS identities or GSM identities. These should be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under <http://webapp.etsi.org/key/queryform.asp>.

Contents

Intellectual Property Rights	2
Foreword.....	2
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.....	8
6 Sequence diagrams	8
6.1 Send picture.....	8
7 XML schema data type definition	9
7.1 DeliveryStatus enumeration	9
7.2 MessagePriority enumeration.....	10
7.3 DeliveryInformation structure	10
7.4 MessageReference structure	10
7.5 MessageURI structure	10
8 Web Service interface definition	10
8.1 Interface: SendMessage	10
8.1.1 Operation: SendMessage	11
8.1.1.1 Input message: SendMessageRequest	11
8.1.1.2 Output message: SendMessageResponse	11
8.1.1.3 Referenced faults.....	11
8.1.2 Operation: GetMessageDeliveryStatus	11
8.1.2.1 Input message: GetMessageDeliveryStatusRequest.....	12
8.1.2.2 Output message: GetMessageDeliveryStatusResponse	12
8.1.2.3 Referenced faults.....	12
8.2 Interface: ReceiveMessage	12
8.2.1 Operation: GetReceivedMessages	12
8.2.1.1 Input message: GetReceivedMessagesRequest.....	12
8.2.1.2 Output message: GetReceivedMessagesResponse	12
8.2.1.3 Referenced faults.....	12
8.2.2 Operation: GetMessageURIs	13
8.2.2.1 Input message: GetMessageURIsRequest.....	13
8.2.2.2 Output message: GetMessageURIsResponse.....	13
8.2.2.3 Referenced faults.....	13
8.2.3 Operation: GetMessage.....	13
8.2.3.1 Input message: GetMessageRequest	13
8.2.3.2 Output message: GetMessageResponse	13
8.2.3.3 Referenced faults.....	13
8.3 Interface: MessageNotification	14
8.3.1 Operation: NotifyMessageReception	14
8.3.1.1 Input message: NotifyMessageReceptionRequest	14
8.3.1.2 Output message: NotifyMessageReceptionResponse	14
8.3.1.3 Referenced faults.....	14
8.4 Interface: MessageNotificationManager	14
8.4.1 Operation: StartMessageNotification.....	14
8.4.1.1 Input message: StartMessageNotificationRequest	15

8.4.1.2	Output message: StartMessageNotificationResponse	15
8.4.1.3	Referenced Faults	15
8.4.2	Operation: StopMessageNotification	15
8.4.2.1	Input message: StopMessageNotificationRequest	15
8.4.2.2	Output message: StopMessageNotificationResponse	15
8.4.2.3	Referenced Faults	15
9	Fault definitions	16
9.1	ServiceException	16
9.1.1	Void	16
10	Service policies	16
Annex A (normative):	WSDL for Multimedia Messaging	17
Annex B (informative):	Change history	18
History		19

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 5 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: "Overview and common data definitions";

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".

1 Scope

The present document is Part 5 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 Multimedia Messaging Web Service aspects of the interface. All aspects of the Multimedia 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".

[5] W3C Recommendation (2 May 2001): "XML Schema Part 2: Datatypes".

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] W3C Note (11 December 2001): "SOAP Messages with Attachments".

NOTE: Available at <http://www.w3.org/TR/SOAP-attachments>.

[8] 3GPP TS 23.140 "Multimedia Messaging Service (MMS); Functional description; Stage 2".

[9] RFC2822: "Internet Message Format".

NOTE: Available at <http://www.ietf.org/rfc/rfc2822.txt>

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].

Whitespace: see definition for CFWS as defined in RFC2822 [9].

3.2 Abbreviations

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

EMS	Enhanced Messaging Service
IM	Instant Messaging
MMS	Multimedia Messaging Service
MMS-C	Multimedia Messaging Service - Centre
SMS	Short Message Service

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 7) and Multimedia Messaging (this part), which provides generic messaging features (including SMS).

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.

Figure 1 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.

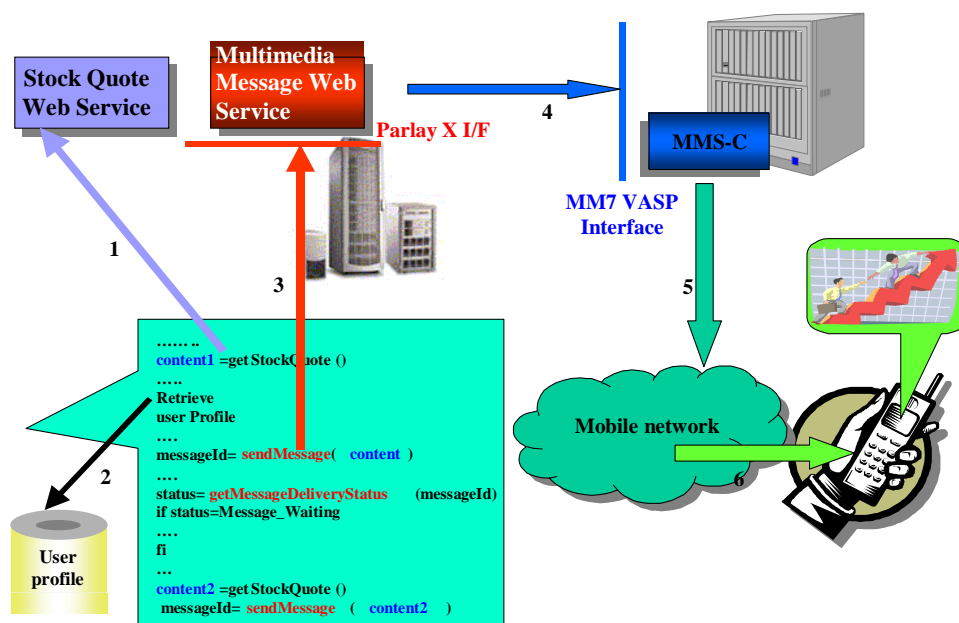


Figure 1: Multimedia Messaging Scenario

5 Namespaces

The SendMessage interface uses the namespace:

www.csapi.org/wsd/parlayx/multimedia_messaging/send/v2_0

The ReceiveMessage interface uses the namespace:

www.csapi.org/wsd/parlayx/multimedia_messaging/receive/v2_0

The MessageNotification interface uses the namespace:

www.csapi.org/wsd/parlayx/multimedia_messaging/notification/v2_0

The data types are defined in the namespace:

www.csapi.org/schema/parlayx/multimedia_messaging/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 picture

With the advent of picture capable phones, the exchange of photos to mobile phones is becoming more common place. This sequence diagram shows an application where a person can send a picture from an online photo album to a mobile phone.

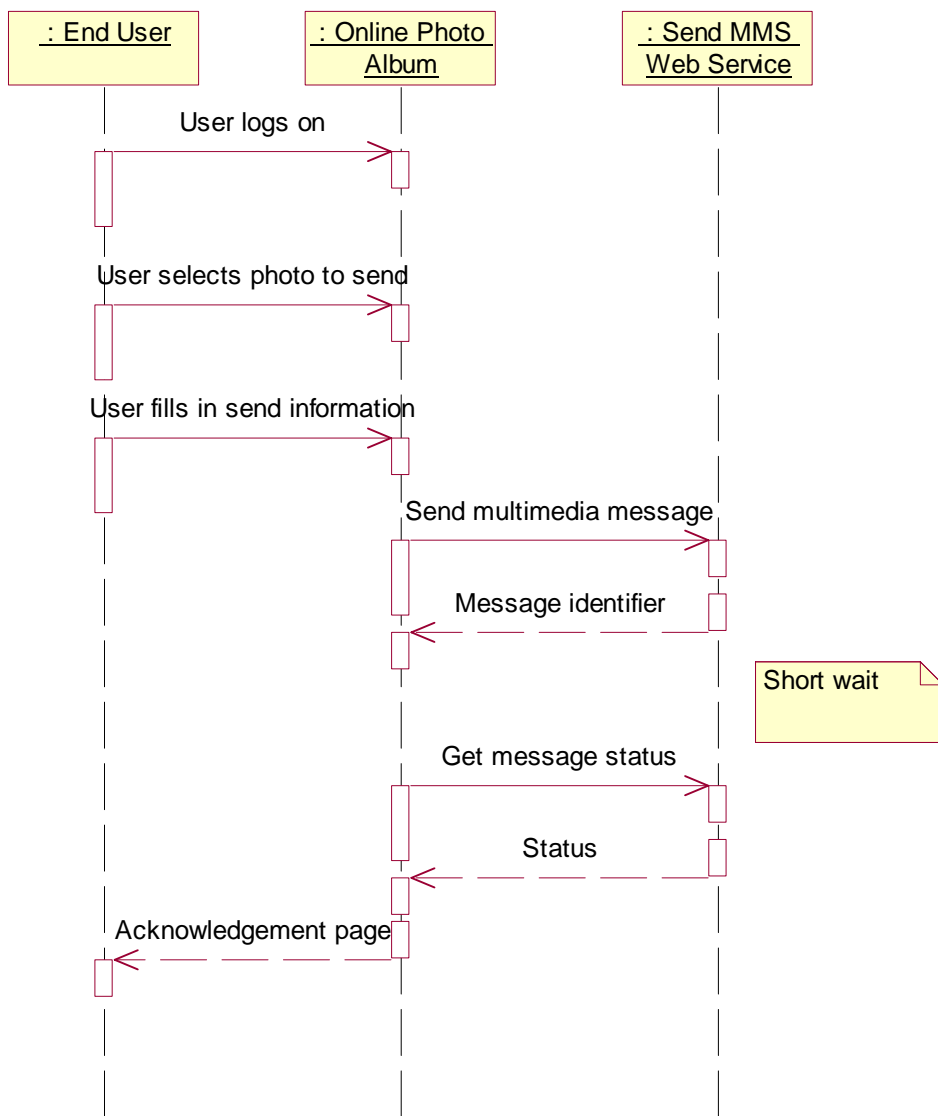


Figure 2

7 XML schema data type definition

7.1 DeliveryStatus enumeration

List of delivery status values.

Enumeration	Description
Delivered	Successful delivery.
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.

7.2 MessagePriority enumeration

List of delivery priority values.

Enumeration	Description
Default	Default message priority
Low	Low message priority
Normal	Normal message priority
High	High message priority

7.3 DeliveryInformation structure

Delivery status information.

Element name	Element type	Description
address	xsd:anyURI	Address associated with the delivery status. The address field is coded as a URI.
deliveryStatus	DeliveryStatus	Parameter indicating the delivery status.

7.4 MessageReference structure

Message information.

Element name	Element type	Description
messageIdentifier	xsd:string	OPTIONAL: If present, contains a reference to a message stored in the Parlay X gateway. If the message is pure text, this parameter is not present.
messageServiceActivationNumber	xsd:string	Number associated with the invoked Message service, i.e. the destination address used by the terminal to send the message.
senderAddress	xsd:anyURI	Indicates message sender address.
subject	xsd:string	OPTIONAL: If present, indicates the subject of the received message. This parameter will not be used for SMS services.
priority	MessagePriority	The priority of the message: default is Normal.
message	xsd:string	OPTIONAL: If present, then the messageIdentifier is not present and this parameter contains the whole message. The type of the message is always pure ASCII text in this case. The message will not be stored in the Parlay X gateway.

7.5 MessageURI structure

Message location information.

Element name	Element type	Description
bodyText	xsd:string	Contains the message body if it is encoded as ASCII text.
fileReferences	xsd:anyURI [0..unbounded]	This is an array of URI references to all the attachments in the Multimedia message. These are URIs to different files, e.g. GIF pictures or pure text files.

8 Web Service interface definition

8.1 Interface: SendMessage

Operations to send messages and check status on sent messages.

8.1.1 Operation: SendMessage

Request to send a Message to a set of destination addresses, returning a **requestIdentifier** to identify the message. The **requestIdentifier** can subsequently be used by the application to poll for the message status, i.e. using **getMessageDeliveryStatus** to see if the message has been delivered or not. The content is sent as a Attachment as specified in SOAP Messages with Attachments [7].

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.

8.1.1.1 Input message: SendMessageRequest

Part name	Part type	Description
Addresses	xsd:anyURI [0..unbounded]	Destination addresses for the Message.
SenderAddress	xsd:string	Message sender address. This parameter is not allowed for all 3 rd party providers. Parlay X server needs to handle this according to a SLA for the specific application and its use can therefore result in a PolicyException. (optional)
Subject	xsd:string	Message subject. If mapped to SMS this parameter will be used as the senderAddress, even if a separate senderAddress is provided. (optional)
Priority	MessagePriority	Priority of the message. If not present, the network will assign a priority based on an operator policy. (optional)
Charging	Common:ChargingInformation	Charging to apply to this message. (optional)

NOTE: The input message may also contain attachments, with appropriate content as defined by SOAP Messages with Attachments [7].

8.1.1.2 Output message: SendMessageResponse

Part name	Part type	Description
RequestIdentifier	xsd:string	It is a correlation identifier that is used in a getMessageDeliveryStatus message invocation, i.e. to poll for the delivery status of all of the sent Messages.

8.1.1.3 Referenced faults

ServiceException from [6]:

- SVC0001 - Service error.
- SVC0002 - Invalid input value.
- SVC0004 - No valid addresses.
- SVC0006 - Invalid group.

PolicyException from [6]:

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

8.1.2 Operation: GetMessageDeliveryStatus

This is a poll method used by the application to retrieve delivery status for each message sent as a result of a previous **sendMessage** message invocation. The **requestIdentifier** parameter identifies this previous message invocation.

8.1.2.1 Input message: GetMessageDeliveryStatusRequest

Part name	Part type	Description
RequestIdentifier	xsd:string	Identifier related to the delivery status request.

8.1.2.2 Output message: GetMessageDeliveryStatusResponse

Part name	Part type	Description
DeliveryStatus	DeliveryInformation [0..unbounded]	It is an array of status of the messages that were previously sent. Each array element represents a sent message: i.e. its destination address and its delivery status.

8.1.2.3 Referenced faults

ServiceException from [6]:

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

PolicyException from [6]:

- POL0001 - Policy error.

8.2 Interface: ReceiveMessage

Operations to retrieve messages that have been received.

8.2.1 Operation: GetReceivedMessages

This method enables the application to poll for new messages associated with a specific **registrationIdentifier**. If the **registrationIdentifier** is not specified, the Parlay X server will return references to all messages sent to the application. The process of binding different **registrationIdentifier** parameters to applications is an off-line process. The Parlay X gateway shall not allow an application to poll for messages using **registrationIdentifier** parameters that are not associated with the application. The priority parameter may be used by the application to retrieve references to higher priority messages, e.g. if Normal is chosen only references to high priority and normal priority messages are returned. If the priority parameter is omitted all message references are returned.

8.2.1.1 Input message: GetReceivedMessagesRequest

Part name	Part type	Description
RegistrationIdentifier	xsd:string	Identifies the off-line provisioning step that enables the application to receive notification of Message reception according to specified criteria.
Priority	MessagePriority	OPTIONAL. The priority of the messages to poll from the Parlay X gateway. All messages of the specified priority and higher will be retrieved. If not specified, all messages shall be returned, i.e. the same as specifying Low.

8.2.1.2 Output message: GetReceivedMessagesResponse

Part name	Part type	Description
Messages	MessageReference [0..unbounded]	It contains an array of messages received according to the specified filter of registrationIdentifier and priority .

8.2.1.3 Referenced faults

ServiceException from [6]:

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

PolicyException from [6]:

- POL0001 - Policy error.

8.2.2 Operation: GetMessageURIs

This method will read the different parts of the message, create local files in the Parlay Gateway and return URI references to them. The application can then simply read each file or just have them presented as links to the end-user. The URIs to the files will be active for an agreed time.

8.2.2.1 Input message: GetMessageURIsRequest

Part name	Part type	Description
MessageRefIdentifier	xsd:string	The identity of the message to retrieve.

8.2.2.2 Output message: GetMessageURIsResponse

Part name	Part type	Description
Message	MessageURI	It contains the complete message, i.e. the textual part of the message, if such exists, and a list of file references for the message attachments, if any.

8.2.2.3 Referenced faults

ServiceException from [6]:

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

PolicyException from [6]:

- POL0001 - Policy error.

8.2.3 Operation: GetMessage

This method will read the whole message. The data is returned as an attachment, as defined in SOAP Messages with Attachments [7], in the return message.

8.2.3.1 Input message: GetMessageRequest

Part name	Part type	Description
MessageRefIdentifier	String	The identity of the message

8.2.3.2 Output message: GetMessageResponse

Part name	Part type	Description
None		

8.2.3.3 Referenced faults

ServiceException from [6]:

- SVC0001 - Service error.

- SVC0002 - Invalid input value.

PolicyException from [6]:

- POL0001 - Policy error.

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.

8.3.1.1 Input message: NotifyMessageReceptionRequest

Part name	Part type	Description
correlator	xsd:string	Correlator provided in request to set up this notification
Message	MessageReference	This parameter contains all the information associated with the received message.

8.3.1.2 Output message: NotifyMessageReceptionResponse

Part name	Part type	Description
None		

8.3.1.3 Referenced faults

None.

8.4 Interface: MessageNotificationManager

The multimedia message notification manager enables applications to set up and tear down notifications for multimedia messages online. The means to provision notifications offline are not specified here.

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 overlaps then SVC0008 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, as defined as the initial characters after discarding any leading Whitespace and ending with a Whitespace or end of the string. The matching shall be case-insensitive. If the subject of the multimedia message is present it shall be used as the string, if not the string is defined as the first plain/text part of the content (see 3GPP TS 23.140 [8]).

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
- SVC0008 – 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
Correlator	xsd:string	Correlator of request to end

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

9 Fault definitions

9.1 ServiceException

9.1.1 Void

The fault code (SVC0230) is reserved and shall not be used.

10 Service policies

Service policies for this service

Name	Type	Description
GroupSupport	xsd:Boolean	Groups may be included with addresses
NestedGroupSupport	xsd:Boolean	Are nested groups supported in group definitions
ChargingSupported	xsd:Boolean	Charging supported for send message operation

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 29199-05-620-doclit.zip) which accompanies the present document.

Annex B (informative): Change history

Change history							
Date	TSG #	TSG Doc.	CR	Rev	Subject/Comment	Old	New
Dec 2003	CN_21	NP-030552	--	--	Submitted to CN#22 for Information	1.0.0	
Jan 2004	--	--	--	--	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: px0326rpcenc.zip px0326rpclit.zip	1.0.1	
Jun 2004	CN_24	NP-040274	--	--	Split into multi-part specification. 29.199-0n, for n=1,2...9. Submitted to CN#24 for Information	1.0.3	
Sep 2004	CN_25	NP-040360	--	--	Draft v200 submitted to TSG CN#25 for Approval.	2.0.0	6.0.0
Dec 2004	CN_26	NP-040487	001	--	Add MessageNotificationManager interface to PXWS Multimedia-Messaging	6.0.0	6.1.0
Mar 2005	CN_27	NP-050021	002	--	Correct criteria	6.1.0	6.2.0
Mar 2005	CN_27	NP-050021	003	--	Fix StopMessageNotification message part	6.1.0	6.2.0

History

Document history		
V6.1.0	December 2004	Publication
V6.2.0	March 2005	Publication