ETSI TS 129 199-3 V9.2.0 (2012-07)

Digital cellular telecommunications system (Phase 2+);
Universal Mobile Telecommunications System (UMTS);
LTE;
Open Service Access (OSA);
Parlay X web services;
Part 3: Call notification
(3GPP TS 29.199-03 version 9.2.0 Release 9)

Reference
RTS/TSGC-0029199-03v920

Keywords
GSM,LTE,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/chaircor/ETSI_support.asp

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://ipr.etsi.org).

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

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.

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 3 of a multi-part deliverable covering the 3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; 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"
Part 15: "Message Broadcast"
Part 16: "Geocoding"
Part 17: "Application driven Quality of Service (QoS)"
Part 18: "Device Capabilities and Configuration"
Part 19: "Multimedia streaming control"
Part 20: "Multimedia multicast session management"

Part 21: "Content Management"
Part 22: "Policy"

1 Scope

The present document is Part 3 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 Call Notification Web Service aspects of the interface. All aspects of the Call Notification 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 CT 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".

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.

3.2 Abbreviations

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

4 Detailed service description

Currently, in order to determine the handling of a subscriber initiated call in telecommunication networks we have to write applications using specific protocols to access Call Control functions provided by network elements. This approach requires a high degree of network expertise. We can also use the OSA gateway approach, invoking standard interfaces to gain access to call control capabilities, but these interfaces are usually perceived to be quite complex by application IT developers. Developers must have advanced telecommunication skills to use Call Control OSA interfaces.

In this clause we will describe a Parlay X Web Service, Call Notification, for handling calls initiated by a subscriber in the network. A (third party) application determines how the call should be treated. The overall scope of this Web Service is to provide simple functions to application developers to determine how a call should be treated. It is possible to request to end the call, continue the call or re-route the call. Optionally, it is also possible to request the media type(s) when the action is to re-route the call. It provides, for example, the capability to route a call to an IVR in order to play a video stream to the calling subscriber. A service policy determines if multimedia application control is supported.

The media types used in the call can be retrieved using **getMediaForParticipant** or **getMediaForCall** in the Audio Call web service.

Using the Web Services, application developers can perform simple handling of network-initiated calls without specific Telco knowledge.

Examples of usage include the following.

Incoming call handling: A subscriber receives a call while he is logged-on to the Internet. Since this occupies his telephone connection, he is regarded as busy by the network. The subscriber has an application that is invoked when somebody tries to call him while he is busy. The application provides the subscriber with a list of choices on how to handle the call (e.g. route the call to voicemail or other media server, redirect the call to a secretary, reject the call). Based on the response of the subscriber the call is handled in the network. Alternatively, the call is re-routed or released depending on the preferences of the subscriber and some context information (e.g. based on the status or location of the subscriber).

Service numbers: An application is triggered whenever a certain service number is dialled. This number is used to connect the caller to one of the maintenance personnel. The application redirects the call to the appropriate maintenance person based on, e.g. calling party number, time, location and availability of the maintenance personnel.

SMS notification of missed calls: An application offers the subscriber the possibility to be notified via SMS whenever he misses a call. The application registers to be notified when calls to its subscribers encounter busy, no-answer or not-reachable. The application does not influence the call treatment, but sends an SMS containing the calling party number, the time and reason why the call was missed.

MediaInteraction: An application is provided information regarding the start of media stream to an end user, the termination of a media stream that the end user is watching, and other media events, e.g. the end-user pausing playback of a media stream. For example, starting to stream a video to an end user, the end user pausing the ongoing video stream and the ending of the video stream.

5 Namespaces

The CallDirection interface uses the namespace:

http://www.csapi.org/wsdl/parlayx/call_direction/v4_2

The CallNotification interface uses the namespace:

http://www.csapi.org/wsdl/parlayx/call_notification/v4_2

The data types are defined in the namespace:

http://www.csapi.org/schema/parlayx/call_notification/v4_2

The CallNotificationManager interface uses the namespace:

http://www.csapi.org/wsdl/parlayx/call_notification/notification_manager/v4_2

The CallDirectionNotificationManager interface uses the namespace:

http://www.csapi.org/wsdl/parlayx/call_direction/notification_manager/v4_2

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 SMS notification of a missed call

Showing the use of the Call Notification and Short Messaging Web Services, an SMS is sent to a person who misses a call (no answer). This sequence assumes that the provisioning of the "no answer" call notification has occurred independently.

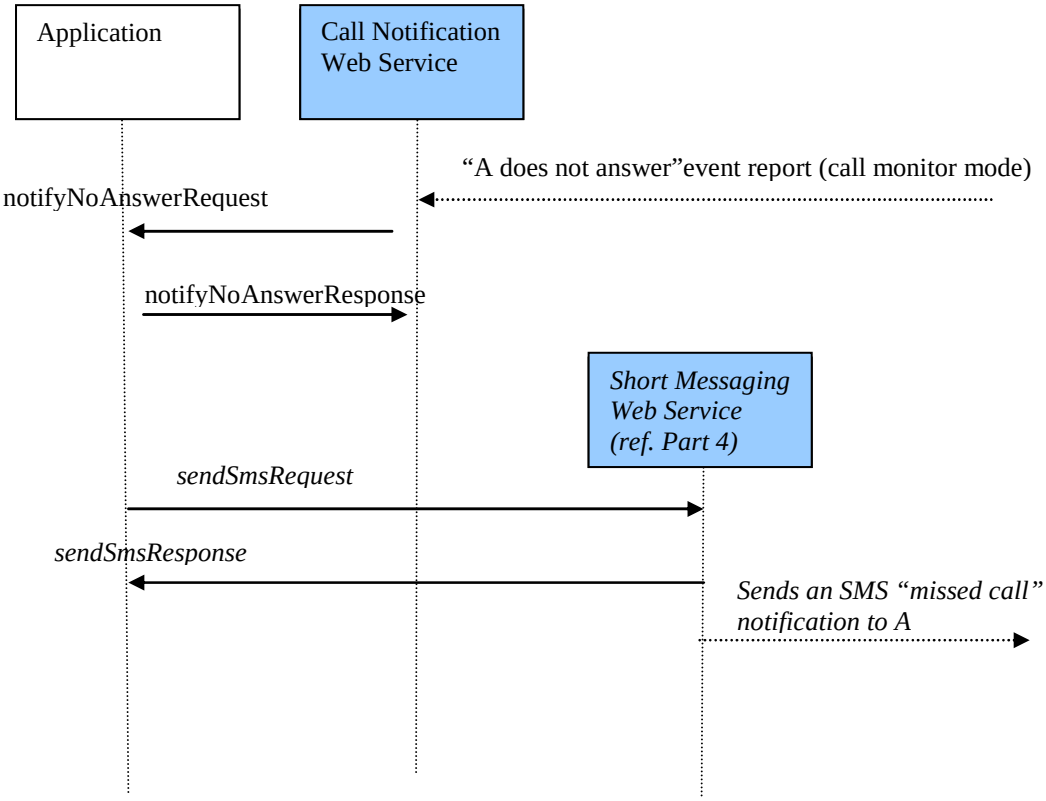

Figure 6.1

6.2 Media Interaction – Collection of Digits from end-user

The application requests the CallNotificationManager to start the process of receiving media notifications. In this example the application requests to receive notifications for the playing of a file and the network collecting digits from the end user.

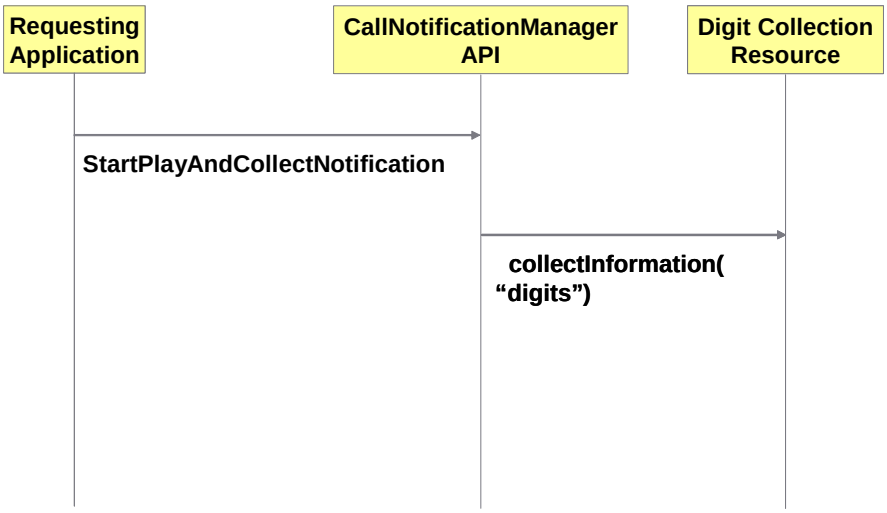

Figure 6.2

6.3 Notification of Media Interaction

In this example the application is being notified of the collection of a digit string that was collected by a digit collection resource.

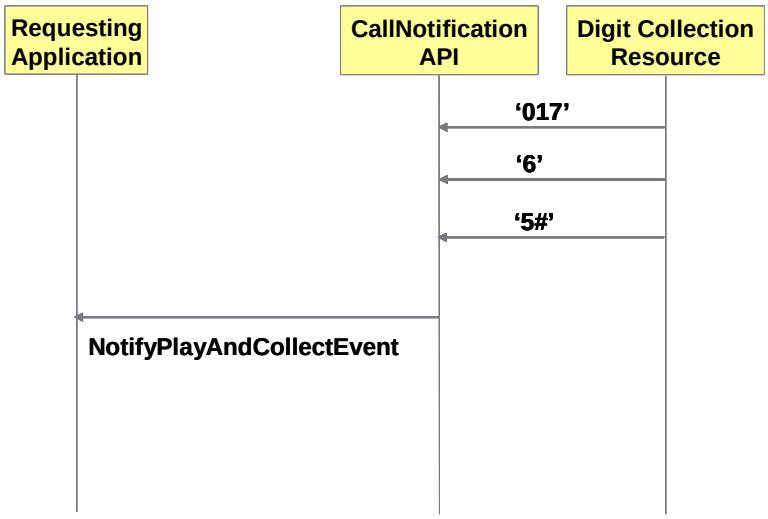

Figure 6.3

6.4 Making ad-hoc conference call after answered

This sequence shows making ad-hoc conference call invoking CN and TPC web service with same session identifier. In this example the application is being notified of the answering the call from user B Then it tries to make conference call among them by means of inviting userC to the existing call between userA and userB with same call session identifier.

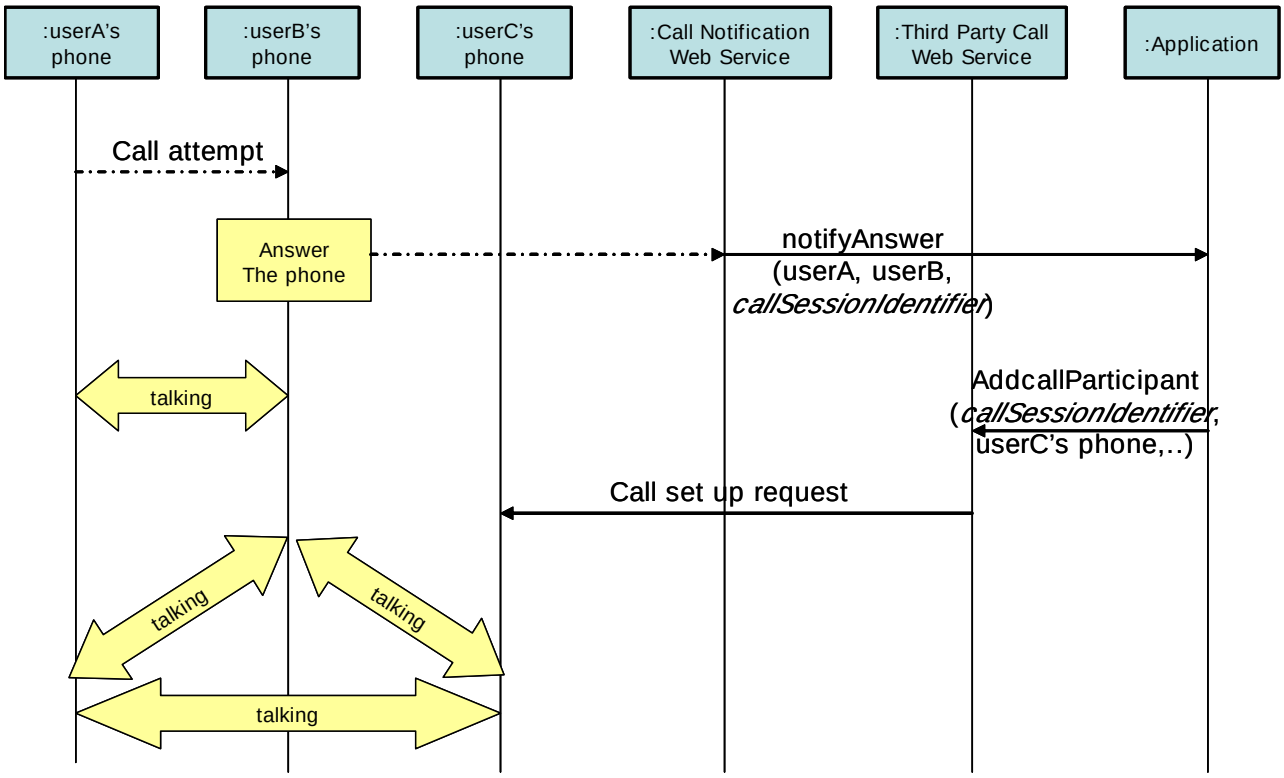

Figure 6.4

7 XML Schema data type definition

7.1 ActionValues enumeration

The **ActionValues** data type is an enumeration with the following values.

Enumeration	Description
Route	Request to (re-)route the call to the address indicated with routingAddress.
Continue	Request to continue the call without any changes. This will result in normal handling of the event in the network.
EndCall	Request to end the call. This will result in termination of the call. The callingParty will receive a tone or announcement.

7.2 Action structure

The **Action** data type is a structure containing the following parameters.

Element name	Element type	Optional	Description
ActionToPerform	ActionValues	No	Indicates the action as described above
RoutingAddress	xsd:anyURI	Yes	The address to be used in case the action indicates 'Route'
Charging	common:ChargingInformation	Yes	Charge to apply to this call session
MediaInfo	common: MediaInfo [0..unbounded]	Yes	The desired media type(s) in case the action indicates 'Route. It identifies one or more media type(s) for the call, i.e. the media type(s) to be applied to the participants in the call session. For each media type the media direction: incoming, outgoing, or bidirectional shall be indicated. It includes the media direction: incoming, outgoing or bidirectional. Only to be used if the action indicates 'Route' An empty array shall have the same meaning as if the parameter is omitted. If the parameter **MediaInfo** is omitted, the media type(s) shall be negotiated by the underlying network.

7.3 CallEvents enumeration

The **CallEvents** data type is an enumeration with the following values.

Enumeration value	Description
Busy	Called party is busy.
NotReachable	Called party is not reachable.
NoAnswer	Called party doesn't answer.
CalledNumber	A call session between two parties, a calling participant and a called participant (called number) is being attempted.
Answer	Called Participant has confirmed (answered) the call.
Disconnected	Called (or calling) party disconnected.

7.4 EventDescription structure

The **EventDescription** data type is a structure containing the following parameters.

Element name	Element type	Optional	Description
CallEvent	CallEvents	No	Indicates the call event as described above
Description	xsd:string	Yes	Optional description to give details about the callEvents (for instance the name of the Participant in case of disconnection) or details about the CallEvent when the callEvent subscribed is service specific.

7.5 AddressDirection enumeration

The **AddressDirection** data type is an enumeration with the following values.

Enumeration value	Description
Called	Called party is considered
Calling	Calling party is considered.

8 Web Service interface definition

8.1 Interface: CallDirection

This subclause describes an initial set of capabilities in terms of message invocations, parameters and data types. The message-based invocations are:

- handleBusy.
- handleNotReachable.
- handleNoAnswer.
- handleCallEvent

  handleCallEvent enables handling for any type of event, thus, making the four handle functions (handleBusy, handleNotReachable, handleNoAnswer, handleCalledNumber) not necessary any more. They could be deprecated in future versions.

These messages are initiated by the Call Notification Web Service (running in a Parlay X Gateway) and invoke an application Web Service(s), as a result of activity in the network. The result of the invocation of a handle<Event> operation is used as an indication on how the call should be handled in the network. The application can only indicate the call handling required prior to the call being established, and cannot keep control over the call after handling the event; every event handling is a separate occurrence.

Note that because the results of the invocations of the application Web Service(s) determine call handling in the network, the names of the methods are prefixed with 'handle', rather than 'notify'. The prefix 'notify' would imply a more asynchronous behaviour, whereas 'handle' shows the synchronous nature of these invocations.

8.1.1 Operation: HandleBusy

The invocation of **handleBusy** requests the application to inform the gateway how to handle the call between two addresses, the **callingParticipant** and the **calledParticipant**, where the **calledParticipant** is busy when the call is received. Optionally, the caller's name is provided. The application returns the **action**, which directs the gateway to perform one of the following actions:

- "Continue", resulting in normal handling of the busy event in the network, e.g. playing of a busy tone to the **callingParticipant.**
- "EndCall", resulting in the call being terminated; the exact tone or announcement that will be played to the **callingParticipant** is operator-specific.
- "Route", resulting in the call being re-routed to a **calledParticipant** specified by the application.

Optionally, in the **action** parameter, the application can also indicate the charging information.

8.1.1.1 Input message: handleBusyRequest

Part name	Part type	Optional	Description
Correlator	xsd:string	No	Correlator provided in request to set up this notification. A unique identifier for the application Web Service which has requested this notification, See also StartCallDirectionNotification
CallingParticipant	xsd:anyURI	No	It contains the address of the caller
CallingParticipantName	xsd:string	Yes	It contains the name of the caller
CalledParticipant	xsd:anyURI	No	It contains the address of the called participant. This participant is busy
CallSessionIdentifier	xsd:string	Yes	Identifies the call session. If provided allows PX applications to avail of additional PX web service features and capabilities that rely upon a callSessionIdentifier.

8.1.1.2 Output message: handleBusyResponse

Part name	Part type	Optional	Description
result	Action	No	It indicates the action to be performed by the gateway

8.1.1.3 Referenced faults

None.

8.1.2 Operation: HandleNotReachable

The invocation of **handleNotReachable** requests the application to inform the gateway how to handle the call between two addresses, the **callingParticipant** and the **calledParticipant**, where the **calledParticipant** is not reachable when the call is received. Optionally, the caller's name is provided. The application returns the **action**, which directs the gateway to perform one of the following actions:

- "Continue", resulting in normal handling of the 'not reachable' event in the network, e.g. playing of a busy tone to the **callingParticipant.**
- "EndCall", resulting in the call being terminated; the exact tone or announcement that will be played to the **callingParticipant** is operator-specific.
- "Route", resulting in the call being re-routed to a **calledParticipant** specified by the application.

Optionally, in the **action** parameter, the application can also indicate the charging information.

8.1.2.1 Input message: handleNotReachableRequest

Part name	Part type	Optional	Description
Correlator	xsd:string	No	Correlator provided in request to set up this notification. A unique identifier for the application Web Service which has requested this notification, See also StartCallDirectionNotification
CallingParticipant	xsd:anyURI	No	It contains the address of the caller
CallingParticipantName	xsd:string	Yes	It contains the name of the caller
CalledParticipant	xsd:anyURI	No	It contains the address of the called participant. This participant is not reachable
CallSessionIdentifier	xsd:string	Yes	Identifies the call session. If provided allows PX applications to avail of additional PX web service features and capabilities that rely upon a callSessionIdentifier.

8.1.2.2 Output message: handleNotReachableResponse

Part name	Part type	Optional	Description
result	Action	No	It indicates the action to be performed by the gateway

8.1.2.3 Referenced faults

None.

8.1.3 Operation: HandleNoAnswer

The invocation of **handleNoAnswer** requests the application to inform the gateway how to handle the call between two addresses, the **callingParticipant** and the **calledParticipant**, where the **calledParticipant** does not answer the received call. Optionally, the caller's name is provided. The application returns the **action**, which directs the gateway to perform one of the following actions:

- "Continue", resulting in normal handling of the 'no answer' event in the network, e.g. playing of a busy tone to the **callingParticipant.**
- "EndCall", resulting in the call being terminated; the exact tone or announcement that will be played to the **callingParticipant** is operator-specific.
- "Route", resulting in the call being re-routed to a **calledParticipant** specified by the application.

Optionally, in the **action** parameter, the application can also indicate the charging information.

8.1.3.1 Input message: handleNoAnswerRequest

Part name	Part type	Optional	Description
Correlator	xsd:string	No	Correlator provided in request to set up this notification. A unique identifier for the application Web Service which has requested this notification, See also StartCallDirectionNotification
CallingParticipant	xsd:anyURI	No	It contains the address of the caller
CallingParticipantName	xsd:string	Yes	It contains the name of the caller
CalledParticipant	xsd:anyURI	No	It contains the address of the called participant. This participant does not answer the call
CallSessionIdentifier	xsd:string	Yes	Identifies the call session. If provided allows PX applications to avail of additional PX web service features and capabilities that rely upon a callSessionIdentifier.

8.1.3.2 Output message: handleNoAnswerResponse

Part name	Part type	Optional	Description
result	Action	No	It indicates the action to be performed by the gateway

8.1.3.3 Referenced faults

None.

8.1.4 Operation: HandleCalledNumber

The invocation of **handleCalledNumber** requests the application to inform the gateway how to handle the call between two addresses, the **callingParticipant** and the **calledParticipant**. The method is invoked when the **callingParticipant** tries to call the **calledParticipant**, but before the network routes the call to the **calledParticipant**. For example, the **calledParticipant** does not have to refer to a real end user, i.e., it could be a service number. Optionally, the caller's name is provided. The application returns the **action**, which directs the gateway to perform one of the following actions:

- "Continue", resulting in normal handling in the network, i.e. the call will be routed to the **calledParticipant** number, as originally dialled.
- "EndCall", resulting in the call being terminated; the exact tone or announcement that will be played to the **callingParticipant** is operator-specific.
- "Route", resulting in the call being re-routed to a **calledParticipant** specified by the application.

Optionally, in the **action** parameter, the application can also indicate the charging information.

8.1.4.1 Input message: handleCalledNumberRequest

Part name	Part type	Optional	Description
Correlator	xsd:string	No	Correlator provided in request to set up this notification. A unique identifier for the application Web Service which has requested this notification, See also StartCallDirectionNotification
CallingParticipant	xsd:anyURI	No	It contains the address of the caller
CallingParticipantName	xsd:string	Yes	It contains the name of the caller
CalledParticipant	xsd:anyURI	No	It contains the address of the called participant
CallSessionIdentifier	xsd:string	Yes	Identifies the call session. If provided allows PX applications to avail of additional PX web service features and capabilities that rely upon a callSessionIdentifier.

8.1.4.2 Output message: handleCalledNumberResponse

Part name	Part type	Optional	Description
result	Action	No	It indicates the action to be performed by the gateway

8.1.4.3 Referenced faults

None.

8.1.5 Operation: HandleCallEvent

The invocation of **handleCallEvent** requests the application to inform the gateway how to handle the call between two addresses, the **callingParticipant** and the **calledParticipant**. The method is invoked when the event subscribed occurs to the targetted **Participant** (calling or called depending on event and subscription).
Optionally, the caller's name is provided. The application returns the **action**, which directs the gateway to perform one of the following actions:

- "Continue", resulting in normal handling in the network, i.e. the call will be routed to the **calledParticipant** number, as originally dialled.
- "EndCall", resulting in the call being terminated; the exact tone or announcement that will be played to the **callingParticipant** is operator-specific.
- "Route", resulting in the call being re-routed to a **calledParticipant** specified by the application.

Optionally, in the **action** parameter, the application can also indicate the charging information.

8.1.5.1 Input message: handleCallEventRequest

Part name	Part type	Optional	Description
Correlator	xsd:string	No	Correlator provided in request to set up this notification. A unique identifier for the application Web Service which has requested this notification, See also StartCallDirectionNotification
EventNotified	xsd:EventDescription	No	Call Event values to generate notification and optional description to give details about the callEvents (for instance the name of the newParticipant) or details about the CallEvent when the callEvent subscribed is service specific.
CallingParticipant	xsd:anyURI	No	It contains the address of the caller
CallingParticipantName	xsd:string	Yes	It contains the name of the caller
CalledParticipant	xsd:anyURI	No	It contains the address of the called participant
CallSessionIdentifier	xsd:string	Yes	Identifies the call session. If provided allows PX applications to avail of additional PX web service features and capabilities that rely upon a callSessionIdentifier.

8.1.5.2 Output message: handleCallEventResponse

Part name	Part type	Optional	Description
result	Action	No	It indicates the action to be performed by the gateway

8.1.5.3 Referenced faults

None.

8.2 Interface: CallNotification

When call events occur in the network, the application may be notified of these events. The application does not have the ability to influence the call, as call processing continues.

Notifications are provided for call attempt, busy, not reachable, answer and no answer events. A generic operation "notifyCallEvent" provides for notifications for new call events. Thus, the operations specific to call attempt, busy, not reachable, answer and no answer are no longer necessary and they could be deprecated in future versions.

8.2.1 Operation: NotifyBusy

A busy notification informs the application that a call between two parties was attempted, but the called party was busy.

8.2.1.1 Input message: NotifyBusyRequest

Part name	Part type	Optional	Description
Correlator	xsd:string	No	Correlator provided in request to set up this notification. A unique identifier for the application Web Service which has requested this notification, See also StartCallNotification
CallingParticipant	xsd:anyURI	No	It contains the address of the caller
CallingParticipantName	xsd:string	Yes	It contains the name of the caller
CalledParticipant	xsd:anyURI	No	It contains the address of the called participant. This participant is busy
CallSessionIdentifier	xsd:string	Yes	Identifies the call session. If provided allows PX applications to avail of additional PX web service features and capabilities that rely upon a callSessionIdentifier.

8.2.1.2 Output message: NotifyBusyResponse

Part name	Part type	Optional	Description
None		No	

8.2.1.3 Referenced faults

None.

8.2.2 Operation: NotifyNotReachable

A not reachable notification informs the application that a call between two parties was attempted, but the called participant was not reachable.

8.2.2.1 Input message: NotifyNotReachableRequest

Part name	Part type	Optional	Description
Correlator	xsd:string	No	Correlator provided in request to set up this notification. A unique identifier for the application Web Service which has requested this notification, See also StartCallNotification
CallingParticipant	xsd:anyURI	No	It contains the address of the caller
CallingParticipantName	xsd:string	Yes	It contains the name of the caller
CalledParticipant	xsd:anyURI	No	It contains the address of the called participant. This participant is not reachable
CallSessionIdentifier	xsd:string	Yes	Identifies the call session. If provided allows PX applications to avail of additional PX web service features and capabilities that rely upon a callSessionIdentifier.

8.2.2.2 Output message: NotifyNotReachableResponse

Part name	Part type	Optional	Description
None		No	

8.2.2.3 Referenced faults

None.

8.2.3 Operation: NotifyNoAnswer

A no answer notification informs the application that a call between two parties was attempted, but the called participant did not answer.

8.2.3.1 Input message: NotifyNoAnswerRequest

Part name	Part type	Optional	Description
Correlator	xsd:string	No	Correlator provided in request to set up this notification. A unique identifier for the application Web Service which has requested this notification, See also StartCallNotification
CallingParticipant	xsd:anyURI	No	It contains the address of the caller
CallingParticipantName	xsd:string	Yes	It contains the name of the caller
CalledParticipant	xsd:anyURI	No	It contains the address of the called participant. This participant did not answer
CallSessionIdentifier	xsd:string	Yes	Identifies the call session. If provided allows PX applications to avail of additional PX web service features and capabilities that rely upon a callSessionIdentifier.

8.2.3.2 Output message: NotifyNoAnswerResponse

Part name	Part type	Optional	Description
None		No	

8.2.3.3 Referenced faults

None.

8.2.4 Operation: NotifyCalledNumber

A called number notification informs the application that a call between two parties is being attempted.

8.2.4.1 Input message: NotifyCalledNumberRequest

Part name	Part type	Optional	Description
Correlator	xsd:string	No	Correlator provided in request to set up this notification. A unique identifier for the application Web Service which has requested this notification, See also StartCallNotification
CallingParticipant	xsd:anyURI	No	It contains the address of the caller
CallingParticipantName	xsd:string	Yes	It contains the name of the caller
CalledParticipant	xsd:anyURI	No	It contains the address of the called participant
CallSessionIdentifier	xsd:string	Yes	Identifies the call session. If provided allows PX applications to avail of additional PX web service features and capabilities that rely upon a callSessionIdentifier.

8.2.4.2 Output message: NotifyCalledNumberResponse

Part name	Part type	Optional	Description
None		No	

8.2.4.3 Referenced faults

None.

8.2.5 Operation: NotifyAnswer

An answer notification informs the application that a call between two parties is in progress.

8.2.5.1 Input message: NotifyAnswerRequest

Part name	Part type	Optional	Description
Correlator	xsd:string	No	Correlator provided in request to set up this notification. A unique identifier for the application Web Service which has requested this notification, See also StartCallNotification
CallingParticipant	xsd:anyURI	No	It contains the address of the caller
CallingParticipantName	xsd:string	Yes	It contains the name of the caller
CalledParticipant	xsd:anyURI	No	It contains the address of the called participant
CallSessionIdentifier	xsd:string	Yes	Identifies the call session. If provided allows PX applications to avail of additional PX web service features and capabilities that rely upon a callSessionIdentifier.

8.2.5.2 Output message: NotifyAnswerResponse

Part name	Part type	Optional	Description
None		No	

8.2.5.2 Referenced faults

None.

8.2.6 Operation : NotifyPlayAndCollectEvent

This operation shall be sent to the application to provide the result of a media interaction (play and collect). The **Correlator** parameter shall relate to the part that matches the information provided by the application in the **Reference** part of the **StartPlayAndCollectNotification** request message and shall allow the application to correlate the two events. The **CallParticipant** part identifies the source of the collected digits. The **MediaInteraction** part shall contain the result of the media interaction, including the digits collected.

8.2.6.1 Input message : NotifyPlayAndCollectEventRequest

Part Name	Part Type	Optional	Description
Correlator	xsd:string	No	The correlator that is associated with the notification registration
CallParticipant	xsd:anyURI	No	The call participant who has generated the Media Interaction Event
MediaInteraction	xsd:string	No	The result of the media interaction

8.2.6.2 Output message : NotifyPlayAndCollectEventResponse

Part Name	Part Type	Optional	Description
None			

8.2.6.3 Reference faults

None.

8.2.7 Operation : NotifyPlayAndRecordEvent

The application shall invoke this operation in order to provide the result of a media interaction (play and record information). The **Correlator** part shall match the information provided by the application in the **Reference** part of the **StartPlayAndCollectNotification** request message. The **CallParticipant** part identifies the end user from whom the digits are collected. The **MediaInteraction** part shall contain the result of the media interaction, including the location of the recorded information.

8.2.7.1 Input message : NotifyPlayAndRecordEventRequest

Part Name	Part Type	Optional	Description
Correlator	xsd:string	No	The correlator that is associated with the notification registration
CallParticipant	xsd:anyURI	No	The call participant who has generated the Media Interaction Event
MediaInteraction	xsd:string	No	The result of the media interaction

8.2.7.2 Output message : NotifyPlayAndRecordEventResponse

Part Name	Part Type	Optional	Description
None			

8.2.7.3 Reference faults

None

8.2.8 Operation: NotifyCallEvent

A CallEvent notification informs the application that the callEvent occurs.

8.2.8.1 Input message: NotifyCallEventRequest

Part name	Part type	Optional	Description
Correlator	xsd:string	No	Correlator provided in request to set up this notification. A unique identifier for the application Web Service which has requested this notification, See also StartCallNotification
EventNotified	xsd:EventDescription	No	Call Event values to generate notification and optional description to give details about the callEvents (for instance the name of the newParticipant) or details about the CallEvent when the callEvent subscribed is service specific.
CallingParticipant	xsd:anyURI	No	It contains the address of the caller
CallingParticipantName	xsd:string	Yes	It contains the name of the caller
CalledParticipant	xsd:anyURI	No	It contains the address of the called participant
CallSessionIdentifier	xsd:string	Yes	Identifies the call session. If provided allows PX applications to avail of additional PX web service features and capabilities that rely upon a callSessionIdentifier.

8.2.8.2 Output message: NotifyCallEventResponse

Part name	Part type	Optional	Description
None		No	

8.2.8.3 Referenced faults

None.

8.3 Interface: CallNotificationManager

The call notification manager enables applications to set up and tear down notifications for calls online.

8.3.1 Operation: StartCallNotification

Start notifications to the application for given called party addresses. The addresses are Address Data items as defined in 3GPP TS 29.199-1 [6].

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.

The ***criteria*** specify the event specific criteria used by application to define the event required. Only events that meet these criteria are notified. If the criteria parameter is not present, all call events will be notified.

The ***AddressDirection*** specifies if the ***Addresses*** above are "Called" or "Calling". This parameter is global (only one value for all addresses in the array). If empty, the address direction is "Called". The address direction value "Calling" applies only to two call event types – "CalledNumber" and "Disconnected". In case other event types are communicated in the "Criteria" parameter in combination with AddressDirection="Calling", a ServiceException (SVC0002) should be returned to the application.

8.3.1.1 Input message: StartCallNotificationRequest

Part name	Part type	Optional	Description
Reference	common:SimpleReference	No	Notification endpoint definition
Addresses	xsd:anyURI [1..unbounded]	No	Party addresses to receive notifications on. The address type/direction is determined by the AddressDirection. If the AddressDirection part is not populated this is for notifications on called party addresses.
Criteria	CallEvents [0..unbounded]	Yes	List of Call Event values to generate notification. All event values are supported on the CallNotification manager interface.
Addressdirection	AddressDirection	Yes	Determine if the address considered is "called" or "calling". It applies for all the addresses.

8.3.1.2 Output message: StartCallNotificationResponse

Part Name	Part Type	Optional	Description
none			

8.3.1.3 Referenced Faults

ServiceException from [6]

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

PolicyException from [6]

- POL0001 – Policy error

8.3.2 Operation: StopCallNotification

The application may end a call notification using this operation

8.3.2.1 Input message: StopCallNotificationRequest

Part name	Part type	Optional	Description
Correlator	xsd:string	No	Correlator of request to end

8.3.2.2 Output message: StopCallNotificationResponse

Part Name	Part Type	Optional	Description
None			

8.3.2.3 Referenced Faults

ServiceException from [6]

- SVC0001 – Service error
- SVC0002 – Invalid input value

PolicyException from [6]

- POL0001 – Policy error

8.3.3 Operation StartPlayAndCollectNotification

The application shall invoke this operation in order to request notifications resulting from media interaction (i.e. the **StartPlayAndCollectInteraction** operation in the Audio Call web service) associated with an existing call session. In the request message, the application shall specify the call session (**CallSessionIdentifier**) and the endpoint (**Reference**) for receiving the notifications.

8.3.3.1 Input message: StartPlayAndCollectNotificationRequest

Part Name	Part Type	Optional	Description
Reference	common:SimpleReference	No	Notification endpoint definition
CallSessionIdentifier	xsd:string	No	Identifies the existing call session.

8.3.3.2 Output message: StartPlayAndCollectNotificationResponse

Part Name	Part Type	Optional	Description
None			

8.3.3.3 Referenced Faults

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

- SVC0001 – Service error
- SVC0002 – Invalid input value
- SVC0005 – Duplicate correlator
- SVC0008 – Overlapping Criteria

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

- POL0001 - Policy error.

8.3.4 Operation StartPlayAndRecordNotification

The application shall invoke this operation in order to request notifications resulting from media interaction (i.e. the **StartPlayAndRecordInteraction** operation in the Audio Call web service) associated with an existing call session. In the request message, the application shall specify the call session (**CallSessionIdentifier**) and the endpoint (**Reference**) for receiving the notifications.

8.3.4.1 Input message: StartPlayAndRecordNotificationRequest

Part Name	Part Type	Optional	Description
Reference	common:SimpleReference	No	Notification endpoint definition
CallSessionIdentifier	xsd:string	No	Identifies the existing call session.

8.3.4.2 Output message: StartPlayAndRecordNotificationResponse

Part Name	Part Type	Optional	Description
None			

8.3.4.3 Referenced Faults

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

- SVC0001 – Service error
- SVC0002 – Invalid input value
- SVC0005 – Duplicate correlator
- SVC0008 – Overlapping Criteria

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

- POL0001 - Policy error.

8.3.5 Operation StopMediaInteractionNotification

The application shall invoke this operation in order to stop receipt of media interaction notifications associated with an existing call session.

8.3.5.1 Input Message StopMediaInteractionNotificationRequest

Part Name	Part Type	Optional	Description
Correlator	xsd:string	No	Correlator of request to end

8.3.5.2 Output Message StopMediaInteractionNotification Response

Part Name	Part Type	Optional	Description
None			

8.3.5.3 Referenced Faults

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

- SVC0001 – Service error
- SVC0002 – Invalid input value

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

- POL0001 - Policy error.

8.4 Interface: CallDirectionManager

The call direction manager enables applications to set up and tear down notifications for calls online.

8.4.1 Operation: StartCallDirectionNotification

Start notifications to the application for given called party addresses. The addresses are Address Data items as defined in 3GPP TS 29.199-1 [6].

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.

The ***criteria*** specify the event specific criteria used by application to define the event required. Only events that meet these criteria are notified. If the criteria parameter is not present, all supported call events for the CallDirectionManager interface (i.e. all except Answer) will be notified.

The ***AddressDirection*** specifies if the ***Addresses*** above are "Called" or "Calling". This parameter is global (only one value for all addresses in the array). If empty, the address direction is "Called". The address direction value "Calling" applies only to two call event types – "CalledNumber" and "Disconnected". In case other event types are communicated in the "Criteria" parameter in combination with AddressDirection="Calling", a ServiceException (SVC0002) should be returned to the application.

8.4.1.1 Input message: StartCallDirectionNotificationRequest

Part name	Part type	Optional	Description
Reference	common:SimpleReference	No	Notification endpoint definition
Addresses	xsd:anyURI [1..unbounded]	No	Party addresses to receive notifications on. The address type/direction is determined by the AddressDirection. If the AddressDirection part is not populated this is for notifications on called party addresses.
Criteria	CallEvents [0..unbounded]	Yes	List of Call Event values to generate notification. All event values, with the exception of answer, are supported on the CallDirection manager interface.
Addressdirection	AddressDirection	Yes	Determine if the address considered is "called" or "calling". It applies for all the addresses.

8.4.1.2 Output message: StartCallDirectionNotificationResponse

Part Name	Part Type	Optional	Description
none			

8.4.1.3 Referenced Faults

ServiceException from [6]

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

PolicyException from [6]

- POL0001 – Policy error

8.4.2 Operation: StopCallDirectionNotification

The application may end a call notification using this operation

8.4.2.1 Input message: StopCallDirectionNotificationRequest

Part name	Part type	Optional	Description
Correlator	xsd:string	No	Correlator of request to end

8.4.2.2 Output message: StopCallDirectionNotificationResponse

Part Name	Part Type	Optional	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

No new faults defined for this service.

10 Service policies

Name	Type	Description
MultimediaSupported	xsd:boolean	Indicates whether multimedia is supported and whether an application can change the media types used in a call.

Annex A (normative): WSDL for call notification

The document/literal WSDL representation of this interface specification is compliant to 3GPP TS 29.199-1 [6] and is contained in text files:

- parlayx_call_direction_interface_4_2.wsdl
- parlayx_call_direction_service_4_2.wsdl
- parlayx_call_direction_manager_interface_4_2.wsdl
- parlayx_call_direction_manager_service_4_2.wsdl
- parlayx_call_notification_interface_4_2.wsdl
- parlayx_call_notification_service_4_1.wsdl
- parlayx_call_notification_manager_interface_4_2.wsdl
- parlayx_call_notification_manager_service_4_2.wsdl
- parlayx_call_notification_types_4_2.xsd

which accompany the present document.

The WSDL files have been verified using the following files:

- 3_wsdl2Java_axis-1_4.bat
- 3_wsdl2Java_axis2-1_4_1.bat

which accompany the present document.

Annex B (informative): Description of Parlay X Web Services Part 3: Call Notification for 3GPP2 cdma2000 networks

This annex is intended to define the OSA Parlay X Web Services Stage 3 interface definitions and it provides the complete OSA specifications. It is an extension of OSA Parlay X Web Services specifications capabilities to enable operation in cdma2000 systems environment. They are in alignment with 3GPP2 Stage 1 requirements and Stage 2 architecture defined in:

[1] 3GPP2 X.S0011-D: “cdma2000 Wireless IP Network Standard ", Version 1.1

[2] 3GPP2 S.R0037-0: "IP Network Architecture Model for cdma2000 Spread Spectrum Systems", Version 3.0

[3] 3GPP2 X.S0013-A: "All-IP Core Network Multimedia Domain"

These requirements are expressed as additions to and/or exclusions from the 3GPP Release 8 specification.
The information given here is to be used by developers in 3GPP2 cdma2000 network architecture to interpret the 3GPP OSA specifications.

B.1 General Exceptions

The terms 3GPP and UMTS are not applicable for the cdma2000 family of standards. Nevertheless these terms are used (3GPP TR 21.905) mostly in the broader sense of "3G Wireless System". If not stated otherwise there are no additions or exclusions required.

CAMEL mappings are not applicable for cdma2000 systems.

B.2 Specific Exceptions

B.2.1 Clause 1: Scope

There are no additions or exclusions.

B.2.2 Clause 2: References

There are no additions or exclusions.

B.2.3 Clause 3: Definitions and abbreviations

There are no additions or exclusions.

B.2.4 Clause 4: Detailed service description

There are no additions or exclusions.

B.2.5 Clause 5: Namespaces

There are no additions or exclusions.

B.2.6 Clause 6: Sequence diagrams

There are no additions or exclusions.

B.2.7 Clause 7: XML Schema data type definition

There are no additions or exclusions.

B.2.8 Clause 8: Web Service interface definition

There are no additions or exclusions.

B.2.9 Clause 9: Fault definitions

There are no additions or exclusions.

B.2.10 Clause 10: Service policies

There are no additions or exclusions.

B.2.11 Annex A (normative): WSDL for call notification

There are no additions or exclusions.

Annex C (informative): Change history

Change history								
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Cat**	**Old**	**New**
Dec 2006	CT_34	CP-060589	0012	--	Correction to CN WSDL files	F	6.3.0	6.4.0
Dec 2006	CT_34	CP-060589	0013	--	Clarify that events are associated with called party address	F	6.3.0	6.4.0
Dec 2006	CT_34	CP-060599	0014	--	Add Extended Call Control for Parlay X	C	6.4.0	7.0.0
Mar 2007	CT_35	CP-070045	0016	-	Add OSA Parlay Web Services support for 3GPP2 networks	A	7.0.0	7.1.0
Mar 2007	CT_35	CP-070048	0018	--	Add support for multimedia in Call Notification	B	7.0.0	7.1.0
Mar 2007	CT_35	CP-070048	0019	--	Add support for Media Interaction	B	7.0.0	7.1.0
Mar 2007	--	--	--	--	Editorial: Aligned 5 Namespaces	--	7.1.0	7.1.1
Jun 2007	CT_36	CP-070346	0021	--	Refine Call Notification Web Service	F	7.1.1	7.2.0
Sep 2007	CT_37	CP-070641	0022	--	Add service scenario to Call Notification web service	C	7.2.0	8.0.0
Dec 2007	CP-38	CP-070714	0023	1	Add a new call event in the CallEvents enumeration: Disconnected	C	8.0.0	8.1.0
Dec 2007	CP-38	CP-070714	0024	1	Add subscription possibilities : calling trigger	C	8.0.0	8.1.0
Dec 2007	CP-38	CP-070714	0025	1	Add generic notification (and "handle") functions for new "callEvents"	B	8.0.0	8.1.0
Sep 2009	CP-45	CP-090591	0026		Completion of Parlay X Part 3 – Call Notification for Release 8	F	8.1.0	8.2.0
2009-12	-	-	-	-	Update to Rel-9 version (MCC)		8.2.0	9.0.0
Jun 2011	CP-52	CP-110491	0028		Missing AddressDirection data structure definition	A	9.0.0	9.1.0
Jun 2012	CP-56	CP-120379	0030		Missing operations and data types in WSDL file structure	A	9.1.0	9.2.0

History

Document history		
V9.0.0	January 2010	Publication
V9.1.0	June 2011	Publication
V9.2.0	July 2012	Publication