Final draft ETSI ES 202 504-3 V1.1.1 (2008-02)

ETSI Standard

Open Service Access (OSA); Parlay X Web Services; Part 3: Call Notification (Parlay X 3)

Reference

DES/TISPAN-01034-3-OSA

Keywords

API, OSA, service

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

Copyright Notification

Contents

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 ETSI Standard (ES) has been produced by ETSI Technical Committee Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN), and is now submitted for the ETSI standards Membership Approval Procedure.

The present document is part 3 of a multi-part deliverable covering 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".

The present document has been defined jointly between ETSI, The Parlay Group (http://www.parlay.org) and the 3GPP.

The present document forms part of the Parlay X 3.0 set of specifications.

The present document is equivalent to 3GPP TS 29.199-03 V7.2.0 (Release 7).

1 Scope

The present document is part 3 of the Stage 3 Parlay X 3 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 present document specifies the Call Notification Web Service. The following are defined here:

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

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific.

- For a specific reference, subsequent revisions do not apply.
- Non-specific reference may be made only to a complete document or a part thereof and only in the following cases:
  - if it is accepted that it will be possible to use all future changes of the referenced document for the purposes of the referring document;
  - for informative references.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

For online referenced documents, information sufficient to identify and locate the source shall be provided. Preferably, the primary source of the referenced document should be cited, in order to ensure traceability. Furthermore, the reference should, as far as possible, remain valid for the expected life of the document. The reference shall include the method of access to the referenced document and the full network address, with the same punctuation and use of upper case and lower case letters.

NOTE: While any hyperlinks included in this clause were valid at the time of publication ETSI cannot guarantee their long term validity.

2.1 Normative references

The following referenced documents are indispensable for the application of the present document. For dated references, only the edition cited applies. For non-specific references, the latest edition of the referenced document (including any amendments) applies.

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

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

[2] ETSI ES 202 504-1: "Open Service Access (OSA); Parlay X Web Services; Part 1: Common (Parlay X 3)".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in ES 202 504-1 [2] apply.

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in ES 202 504-1 [2] 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 the **getMediaForParticipant** or **getMediaForCall** operations of 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 a 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/v3_2

The CallDirectionNotificationManager interface uses the namespace:

http://www.csapi.org/wsdl/parlayx/call_direction/notification_manager/v3_1

The CallNotification interface uses the namespace:

http://www.csapi.org/wsdl/parlayx/call_notification/v3_2

The CallNotificationManager interface uses the namespace:

http://www.csapi.org/wsdl/parlayx/call_notification/notification_manager/v3_2

The data types are defined in the namespace:

http://www.csapi.org/schema/parlayx/call_notification/v3_1

The "xsd" namespace is used in the present document to refer to the XML Schema data types defined in XML Schema [1]. 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 CallNotification 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.

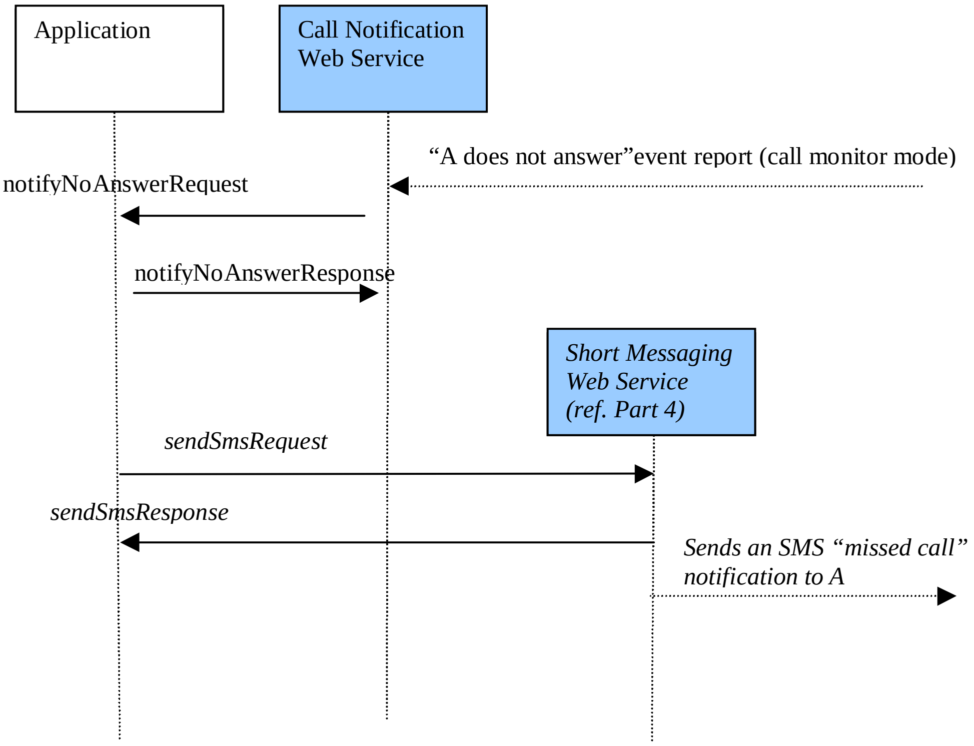
Application
Call Notification Web Service
"A does not answer"event report (call monitor mode)
notifyNoAnswerRequest
notifyNoAnswerResponse
Short Messaging Web Service (ref. Part 4)
sendSmsRequest
sendSmsResponse
Sends an SMS "missed call" notification to A

Figure 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.

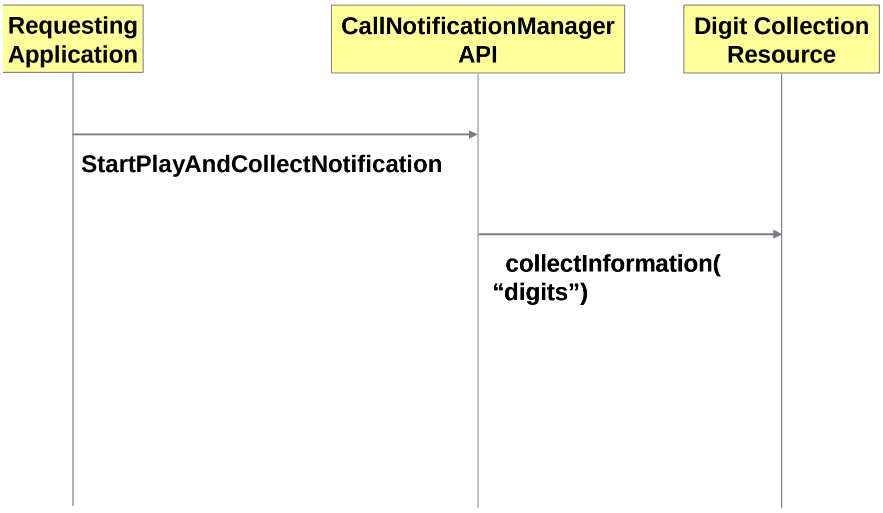
Requesting Application
CallNotificationManager API
Digit Collection Resource
StartPlayAndCollectNotification
collectInformation("digits")

Figure 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.

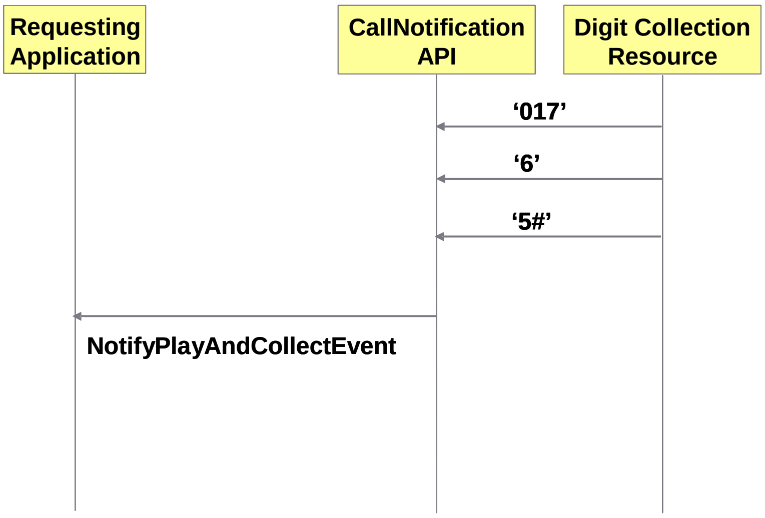

Figure 3

7 XML Schema data type definition

7.1 ActionValues enumeration

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

Enumeration value	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:Charging Information	Yes	Charge to apply to this call session
mediaInfo	common:MediaInfo [0..unbounded]	Yes	Indicates the desired media type(s) for the case where **actionToPerform**=**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. An empty array shall have the same meaning as if the parameter is omitted: i.e. 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 does not 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.

8 Web Service interface definition

8.1 Interface: CallDirection

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

- handleBusy.
- handleNotReachable.
- handleNoAnswer.
- handleCalledNumber.

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 **calledParticipanty**, 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 the request to set up this notification. A unique identifier for the application Web Service which has requested this notification. See also the **startCallDirectionNotification** operation.
callingParticipant	xsd:anyURI	No	It contains the address of the calle.
callingParticipant Name	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
callSession Identifier	xsd:string	Yes	Identifies the call session. If provided, it allows applications to avail of additional Parlay X 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 the request to set up this notification. A unique identifier for the application Web Service which has requested this notification. See also the **startCallDirectionNotification** operation.
callingParticipant	xsd:anyURI	No	It contains the address of the caller.
callingParticipant Name	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.
callSession Identifier	xsd:string	Yes	Identifies the call session. If provided, it allows applications to avail of additional Parlay X 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 the request to set up this notification. A unique identifier for the application Web Service which has requested this notification. See also the **startCallDirectionNotification** operation.
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.
callSession Identifier	xsd:string	Yes	Identifies the call session. If provided, it allows applications to avail of additional Parlay X 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 the request to set up this notification. A unique identifier for the application Web Service which has requested this notification. See also the **startCallDirectionNotification** operation.
callingParticipant	xsd:anyURI	No	It contains the address of the caller.
callingParticipant Name	xsd:string	Yes	It contains the name of the caller.
calledParticipant	xsd:anyURI	No	It contains the address of the called participant.
callSession Identifier	xsd:string	Yes	Identifies the call session. If provided, it allows applications to avail of additional Parlay X 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.

8.2 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.

8.2.1 Operation: notifyBusy

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

8.2.1.1 Input message: notifyBusyRequest

Part name	Part type	Optional	Description
correlator	xsd:string	No	Correlator provided in the request to set up this notification. A unique identifier for the application Web Service which has requested this notification. See also the **startCallNotification** operation.
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
callSession Identifier	xsd:string	Yes	Identifies the call session. If provided, it allows applications to avail of additional Parlay X web service features and capabilities that rely upon a **callSessionIdentifier**.

8.2.1.2 Output message: notifyBusyResponse

Part name	Part type	Optional	Description
None			

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 the request to set up this notification. A unique identifier for the application Web Service which has requested this notification. See also the **startCallNotification** operation.
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.
callSession Identifier	xsd:string	Yes	Identifies the call session. If provided, it allows applications to avail of additional Parlay X web service features and capabilities that rely upon a **callSessionIdentifier**.

8.2.2.2 Output message: notifyNotReachableResponse

Part name	Part type	Optional	Description
None			

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 the request to set up this notification. A unique identifier for the application Web Service which has requested this notification. See also the **startCallNotification** operation.
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.
callSession Identifier	xsd:string	Yes	Identifies the call session. If provided, it allows applications to avail of additional Parlay X web service features and capabilities that rely upon a **callSessionIdentifier**.

8.2.3.2 Output message: notifyNoAnswerResponse

Part name	Part type	Optional	Description
None			

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 the request to set up this notification. A unique identifier for the application Web Service which has requested this notification. See also the **startCallNotification** operation.
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
callSession Identifier	xsd:string	Yes	Identifies the call session. If provided, it allows applications to avail of additional Parlay X web service features and capabilities that rely upon a **callSessionIdentifier**.

8.2.4.2 Output message: notifyCalledNumberResponse

Part name	Part type	Optional	Description
None			

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 the request to set up this notification. A unique identifier for the application Web Service which has requested this notification. See also the **startCallNotification** operation.
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
callSession Identifier	xsd:string	Yes	Identifies the call session. If provided, it allows applications to avail of additional Parlay X web service features and capabilities that rely upon a **callSessionIdentifier**.

8.2.5.2 Output message: notifyAnswerResponse

Part name	Part type	Optional	Description
None			

8.2.5.3 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** part shall match the information provided by the application in the **reference** part of the **startPlayAndCollectNotificationRequest** message and shall allow the application to correlate the event with the event registration. 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 Referenced 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 **startPlayAndCollectNotificationRequest** 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 Referenced faults

None.

8.3 Interface: CallDirectionManager

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

8.3.1 Operation: startCallDirectionNotification

This operation initiates notifications to the application for the specified called party **addresses**, which are Address Data items as defined in ES 202 504-1 [2].

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

The **criteria** specifies the event-specific criteria used by application to define the call event(s) required. Only events that meet this 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.

8.3.1.1 Input message: startCallDirectionNotificationRequest

Part name	Part type	Optional	Description
reference	common:SimpleReference	No	Notification endpoint definition.
addresses	xsd:anyURI [1..unbounded]	No	Called party addresses for which to receive notifications.
criteria	CallEvents [0..unbounded]	Yes	Call events, excluding the **Answer** event, for which a notification is required. If not specified, all call events (except **Answer**) are notified.

8.3.1.2 Output message: startCallDirectionNotificationResponse

Part Name	Part Type	Optional	Description
None			

8.3.1.3 Referenced Faults

ServiceException from ES 202 504-1 [2]:

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

PolicyException from ES 202 504-1 [2]:

- POL0001 - Policy error

8.3.2 Operation: stopCallDirectionNotification

The application may end a call direction notification using this operation.

8.3.2.1 Input message: stopCallDirectionNotificationRequest

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

8.3.2.2 Output message: stopCallDirectionNotificationResponse

Part Name	Part Type	Optional	Description
None			

8.3.2.3 Referenced Faults

ServiceException from ES 202 504-1 [2]:

- SVC0001 - Service error
- SVC0002 - Invalid input value

PolicyException from ES 202 504-1 [2]:

- 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 ES 202 504-1 [2]:

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

PolicyException from ES 202 504-1 [2]:

- 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 ES 202 504-1 [2]:

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

PolicyException from ES 202 504-1 [2]:

- 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: stopMediaInteractionNotificationResponse

Part Name	Part Type	Optional	Description
None			

8.3.5.3 Referenced Faults

ServiceException from ES 202 504-1 [2]:

- SVC0001 - Service error
- SVC0002 - Invalid input value

PolicyException from ES 202 504-1 [2]:

- POL0001 - Policy error

8.4 Interface: CallNotificationManager

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

8.4.1 Operation: StartCallNotification

This operation initiates notifications to the application for the specified called party **addresses**, which are Address Data items as defined in ES 202 504-1 [2].

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

The **criteria** specifies the event-specific criteria used by application to define the call event(s) required. Only events that meet this criteria are notified. If the criteria parameter is not present, all call events will be notified.

8.4.1.1 Input message: startCallNotificationRequest

Part name	Part type	Optional	Description
reference	common:SimpleReference	No	Notification endpoint definition.
addresses	xsd:anyURI [1..unbounded]	No	Called party addresses for which to receive notifications.
criteria	CallEvents [0..unbounded]	Yes	Call events, including **Answer**, for which a notification is required. If not specified, all call events are notified.

8.4.1.2 Output message: startCallNotificationResponse

Part Name	Part Type	Optional	Description
None			

8.4.1.3 Referenced Faults

ServiceException from ES 202 504-1 [2]:

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

PolicyException from ES 202 504-1 [2]:

- POL0001 - Policy error

8.4.2 Operation: stopCallNotification

The application may end a call notification using this operation.

8.4.2.1 Input message: stopCallNotificationRequest

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

8.4.2.2 Output message: stopCallNotificationResponse

Part Name	Part Type	Optional	Description
None			

8.4.2.3 Referenced Faults

ServiceException from ES 202 504-1 [2]:

- SVC0001 - Service error
- SVC0002 - Invalid input value

PolicyException from ES 202 504-1 [2]:

- POL0001 - Policy error

9 Fault definitions

No new faults are defined for this service.

10 Service policies

The following service policies are defined for the Call Notification service.

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 ES 202 504-1 [2] and is contained in text files (contained in archive es_20250403v010101m0.zip) which accompany the present document.

Annex B (informative): Bibliography

ETSI TR 121 905: "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Vocabulary for 3GPP Specifications (3GPP TR 21.905)".

History

Document history		
V1.1.1	February 2008	Membership Approval Procedure MV 20080425: 2008-02-26 to 2008-04-25