

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



Reference

DES/TISPAN-01007-03-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/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.

© The Parlay Group 2005.

All rights reserved.

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

Contents

Intellectual Property Rights	5
Foreword.....	5
1 Scope	6
2 References	6
3 Definitions and abbreviations.....	6
3.1 Definitions	6
3.2 Abbreviations	6
4 Detailed service description	7
5 Namespaces.....	7
6 Sequence diagrams	7
6.1 SMS notification of a missed call.....	7
7 XML Schema data type definition	8
7.1 ActionValues enumeration	8
7.2 Action structure	8
8 Web Service interface definition.....	8
8.1 Interface: CallDirection	8
8.1.1 Operation: HandleBusy.....	9
8.1.1.1 Input message: handleBusyRequest	9
8.1.1.2 Output message: handleBusyResponse	9
8.1.1.3 Referenced faults.....	9
8.1.2 Operation: HandleNotReachable	9
8.1.2.1 Input message: handleNotReachableRequest.....	10
8.1.2.2 Output message: handleNotReachableResponse.....	10
8.1.2.3 Referenced faults.....	10
8.1.3 Operation: HandleNoAnswer.....	10
8.1.3.1 Input message: handleNoAnswerRequest	10
8.1.3.2 Output message: handleNoAnswerResponse	10
8.1.3.3 Referenced faults.....	10
8.1.4 Operation: HandleCalledNumber	10
8.1.4.1 Input message: handleCalledNumberRequest	11
8.1.4.2 Output message: handleCalledNumberResponse	11
8.1.4.3 Referenced faults.....	11
8.2 Interface: CallNotification.....	11
8.2.1 Operation: NotifyBusy.....	11
8.2.1.1 Input message: NotifyBusyRequest	11
8.2.1.2 Output message: NotifyBusyResponse	11
8.2.1.3 Referenced faults.....	11
8.2.2 Operation: NotifyNotReachable	11
8.2.2.1 Input message: NotifyNotReachableRequest	12
8.2.2.2 Output message: NotifyNotReachableResponse	12
8.2.2.3 Referenced faults.....	12
8.2.3 Operation: NotifyNoAnswer.....	12
8.2.3.1 Input message: NotifyNoAnswerRequest	12
8.2.3.2 Output message: NotifyNoAnswerResponse	12
8.2.3.3 Referenced faults.....	12
8.2.4 Operation: NotifyCalledNumber	12
8.2.4.1 Input message: NotifyCalledNumberRequest	12
8.2.4.2 Output message: NotifyCalledNumberResponse	12
8.2.4.3 Referenced faults.....	12
9 Fault definitions.....	13

10	Service policies	13
Annex A (normative):	WSDL for Call Notification	14
Annex B (informative):	Bibliography.....	15
History		16

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".

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

The present document is equivalent to 3GPP TS 29.199-03 V6.0.0 (Release 6).

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 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

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 and/or edition number or version number) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.

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

[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 202 391-1: "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 ES 202 391-1 [2] apply.

3.2 Abbreviations

For the purposes of the present document, the abbreviations defined in ES 202 391-1 [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. 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, 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.

5 Namespaces

The Call Notification interface uses the namespace:

`www.csapi.org/wsdl/parlayx/call_notification/v2_0`

The data types are defined in the namespace:

`www.csapi.org/schema/parlayx/call_notification/v2_0`

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 SendSms services, an SMS is sent to a person who misses a call (no answer). This sequence assumes that the provisioning of the call notification has occurred independently.

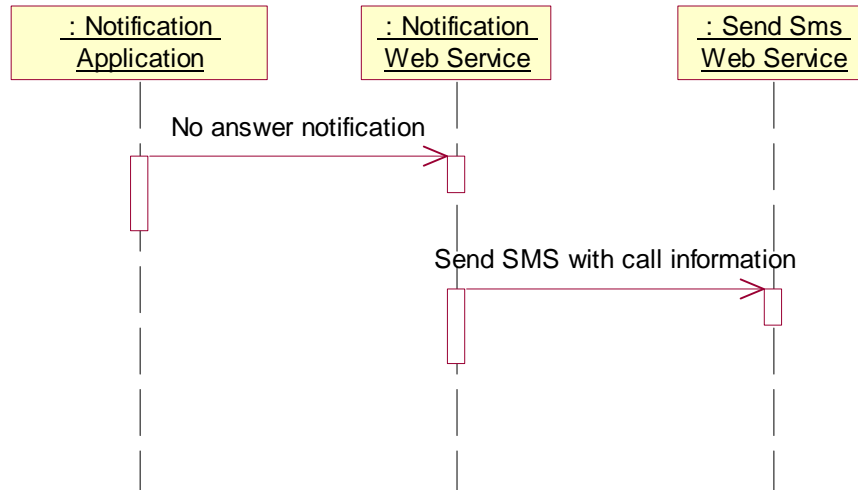


Figure 1

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	Description
ActionToPerform	ActionValues	Indicates the action as described below
RoutingAddress	xsd:anyURI	The address to be used in case the action indicates 'Route'
Charging	common:ChargingInformation	Charge to apply to this 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 not 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.

The criteria for which the application Web Service(s) should be invoked, such as type of events (busy, answer, etc.), a URI to the Web Service and triggered addresses should be provisioned by the operator in an off-line process.

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 **callingParty** and the **calledParty**, where the **calledParty** is busy when the call is received. 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 **callingParty**.
- "EndCall", resulting in the call being terminated; the exact tone or announcement that will be played to the **callingParty** is operator-specific.
- "Route", resulting in the call being re-routed to a **calledParty** 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	Description
CallingParty	xsd:anyURI	It contains the address of the caller
CalledParty	xsd:anyURI	It contains the address of the called party. This party is busy

8.1.1.2 Output message: handleBusyResponse

Part name	Part type	Description
Action	Action	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 **callingParty** and the **calledParty**, where the **calledParty** is not reachable when the call is received. 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 **callingParty**.
- "EndCall", resulting in the call being terminated; the exact tone or announcement that will be played to the **callingParty** is operator-specific.
- "Route", resulting in the call being re-routed to a **calledParty** 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	Description
CallingParty	xsd:anyURI	It contains the address of the caller
CalledParty	xsd:anyURI	It contains the address of the called party. This party is not reachable

8.1.2.2 Output message: handleNotReachableResponse

Part name	Part type	Description
Action	Action	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 **callingParty** and the **calledParty**, where the **calledParty** does not answer the received call. 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 **callingParty**.
- "EndCall", resulting in the call being terminated; the exact tone or announcement that will be played to the **callingParty** is operator-specific.
- "Route", resulting in the call being re-routed to a **calledParty** 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	Description
CallingParty	xsd:anyURI	It contains the address of the caller
CalledParty	xsd:anyURI	It contains the address of the called party. This party does not answer the call

8.1.3.2 Output message: handleNoAnswerResponse

Part name	Part type	Description
Action	Action	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 **callingParty** and the **calledParty**. The method is invoked when the **callingParty** tries to call the **calledParty**, but before the network routes the call to the **calledParty**. For example, the **calledParty** does not have to refer to a real end user, i.e. it could be a service number. 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 **calledParty** number, as originally dialed.

- "EndCall", resulting in the call being terminated; the exact tone or announcement that will be played to the **callingParty** is operator-specific.
- "Route", resulting in the call being re-routed to a **calledParty** 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	Description
CallingParty	xsd:anyURI	It contains the address of the caller
CalledParty	xsd:anyURI	It contains the address of the called party

8.1.4.2 Output message: handleCalledNumberResponse

Part name	Part type	Description
Action	Action	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 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 party was busy.

8.2.1.1 Input message: NotifyBusyRequest

Part name	Part type	Description
CallingParty	xsd:anyURI	It contains the address of the caller
CalledParty	xsd:anyURI	It contains the address of the called party. This party is busy

8.2.1.2 Output message: NotifyBusyResponse

Part name	Part type	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 party was not reachable.

8.2.2.1 Input message: NotifyNotReachableRequest

Part name	Part type	Description
CallingParty	xsd:anyURI	It contains the address of the caller
CalledParty	xsd:anyURI	It contains the address of the called party. This party is not reachable

8.2.2.2 Output message: NotifyNotReachableResponse

Part name	Part type	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 party did not answer.

8.2.3.1 Input message: NotifyNoAnswerRequest

Part name	Part type	Description
CallingParty	xsd:anyURI	It contains the address of the caller
CalledParty	xsd:anyURI	It contains the address of the called party. This party did not answer

8.2.3.2 Output message: NotifyNoAnswerResponse

Part name	Part type	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	Description
CallingParty	xsd:anyURI	It contains the address of the caller
CalledParty	xsd:anyURI	It contains the address of the called party

8.2.4.2 Output message: NotifyCalledNumberResponse

Part name	Part type	Description
None		

8.2.4.3 Referenced faults

None.

9 Fault definitions

No new faults are defined for this service.

10 Service policies

No service policies are defined for this service.

Annex A (normative): WSDL for Call Notification

The document/literal WSDL representation of this interface specification is compliant to ES 202 391-1 [2] and is contained in text files (contained in archive es_20239103v010101m0.zip) which accompany the present document.

Annex B (informative): Bibliography

ETSI TR 121 905: "Universal Mobile Telecommunications System (UMTS); Vocabulary for 3GPP Specifications (3GPP TR 21.905)".

History

Document history		
V1.1.1	January 2005	Membership Approval Procedure MV 20050318: 2005-01-18 to 2005-03-18