INTERNATIONAL STANDARD

ISO/IEC 29341-26-16

First edition
2017-09

Information technology — UPnP Device Architecture —

Part 26-16:
Telephony device control protocol — Level 2 — Presence service

Technologies de l'information — Architecture de dispositif UPnP —

Partie 26-16: Protocole de contrôle de dispositif de téléphonie — Niveau 2 — Service de présence

COPYRIGHT PROTECTED DOCUMENT

CONTENTS

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see http://www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the voluntary nature of Standard, the meaning of the ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword – Supplementary information

ISO/IEC 29341-26-16 was prepared by UPnP Forum and adopted, under the PAS procedure, by joint technical committee ISO/IEC JTC 1, *Information technology*, in parallel with its approval by national bodies of ISO and IEC.

The list of all currently available parts of ISO/IEC 29341 series, under the general title *Information technology — UPnP Device Architecture*, can be found on the ISO web site.

Introduction

ISO and IEC draw attention to the fact that it is claimed that compliance with this document may involve the use of patents as indicated below.

ISO and IEC take no position concerning the evidence, validity and scope of these patent rights. The holders of -these patent rights have assured ISO and IEC that they are willing to negotiate licenses under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statements of the holders of these patent rights are registered with ISO and IEC.

Intel Corporation has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Intel Corporation
Standards Licensing Department
5200 NE Elam Young Parkway
MS: JFS-98
USA – Hillsboro, Oregon 97124

Microsoft Corporation has informed IEC and ISO that it has patent applications or granted patents as listed below:

6101499 / US; 6687755 / US; 6910068 / US; 7130895 / US; 6725281 / US; 7089307 / US; 7069312 / US; 10/783 524 /US

Information may be obtained from:

Microsoft Corporation
One Microsoft Way
USA – Redmond WA 98052

Philips International B.V. has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Philips International B.V. – IP&S
High Tech campus, building 44 3A21
NL – 5656 Eindhoven

NXP B.V. (NL) has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

NXP B.V. (NL)
High Tech campus 60
NL – 5656 AG Eindhoven

Matsushita Electric Industrial Co. Ltd. has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Matsushita Electric Industrial Co. Ltd.
1-3-7 Shiromi, Chuoh-ku
JP – Osaka 540-6139

Hewlett Packard Company has informed IEC and ISO that it has patent applications or granted patents as listed below:

5 956 487 / US; 6 170 007 / US; 6 139 177 / US; 6 529 936 / US; 6 470 339 / US; 6 571 388 / US; 6 205 466 / US

Information may be obtained from:

Hewlett Packard Company
1501 Page Mill Road
USA – Palo Alto, CA 94304

Samsung Electronics Co. Ltd. has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Digital Media Business, Samsung Electronics Co. Ltd.
416 Maetan-3 Dong, Yeongtang-Gu,
KR – Suwon City 443-742

Huawei Technologies Co., Ltd. has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Huawei Technologies Co., Ltd.
Administration Building, Bantian Longgang District
Shenzhen – China 518129

Qualcomm Incorporated has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Qualcomm Incorporated
5775 Morehouse Drive
San Diego, CA – USA 92121

Telecom Italia S.p.A.has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Telecom Italia S.p.A.
Via Reiss Romoli, 274
Turin - Italy 10148

Cisco Systems informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA – USA 95134

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those identified above. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Original UPnP Document

Reference may be made in this document to original UPnP documents. These references are retained in order to maintain consistency between the specifications as published by ISO/IEC and by UPnP Implementers Corporation and later by UPnP Forum. The following table indicates the original UPnP document titles and the corresponding part of ISO/IEC 29341:

UPnP Document Title	ISO/IEC 29341 Part
UPnP Device Architecture 1.0	ISO/IEC 29341-1:2008
UPnP Device Architecture Version 1.0	ISO/IEC 29341-1:2011
UPnP Device Architecture 1.1	ISO/IEC 29341-1-1:2011
UPnP Device Architecture 2.0	ISO/IEC 29341-1-2
UPnP Basic:1 Device	ISO/IEC 29341-2
UPnP AV Architecture:1	ISO/IEC 29341-3-1:2008
UPnP AV Architecture:1	ISO/IEC 29341-3-1:2011
UPnP AVTransport:1 Service	ISO/IEC 29341-3-10
UPnP ConnectionManager:1 Service	ISO/IEC 29341-3-11
UPnP ContentDirectory:1 Service	ISO/IEC 29341-3-12
UPnP RenderingControl:1 Service	ISO/IEC 29341-3-13
UPnP MediaRenderer:1 Device	ISO/IEC 29341-3-2
UPnP MediaRenderer:2 Device	ISO/IEC 29341-3-2:2011
UPnP MediaServer:1 Device	ISO/IEC 29341-3-3
UPnP AVTransport:2 Service	ISO/IEC 29341-4-10:2008
UPnP AVTransport:2 Service	ISO/IEC 29341-4-10:2011
UPnP ConnectionManager:2 Service	ISO/IEC 29341-4-11:2008
UPnP ConnectionManager:2 Service	ISO/IEC 29341-4-11:2011
UPnP ContentDirectory:2 Service	ISO/IEC 29341-4-12
UPnP RenderingControl:2 Service	ISO/IEC 29341-4-13:2008
UPnP RenderingControl:2 Service	ISO/IEC 29341-4-13:2011
UPnP ScheduledRecording:1	ISO/IEC 29341-4-14
UPnP ScheduledRecording:2	ISO/IEC 29341-4-14:2011
UPnP MediaRenderer:2 Device	ISO/IEC 29341-4-2
UPnP MediaServer:2 Device	ISO/IEC 29341-4-3
UPnP AV Datastructure Template:1	ISO/IEC 29341-4-4:2008
UPnP AV Datastructure Template:1	ISO/IEC 29341-4-4:2011
UPnP DigitalSecurityCamera:1 Device	ISO/IEC 29341-5-1
UPnP DigitalSecurityCameraMotionImage:1 Service	ISO/IEC 29341-5-10
UPnP DigitalSecurityCameraSettings:1 Service	ISO/IEC 29341-5-11
UPnP DigitalSecurityCameraStillImage:1 Service	ISO/IEC 29341-5-12
UPnP HVAC_System:1 Device	ISO/IEC 29341-6-1
UPnP ControlValve:1 Service	ISO/IEC 29341-6-10
UPnP HVAC_FanOperatingMode:1 Service	ISO/IEC 29341-6-11
UPnP FanSpeed:1 Service	ISO/IEC 29341-6-12
UPnP HouseStatus:1 Service	ISO/IEC 29341-6-13
UPnP HVAC_SetpointSchedule:1 Service	ISO/IEC 29341-6-14
UPnP TemperatureSensor:1 Service	ISO/IEC 29341-6-15
UPnP TemperatureSetpoint:1 Service	ISO/IEC 29341-6-16

UPnP HVAC_UserOperatingMode:1 Service	ISO/IEC 29341-6-17
UPnP HVAC_ZoneThermostat:1 Device	ISO/IEC 29341-6-2
UPnP BinaryLight:1 Device	ISO/IEC 29341-7-1
UPnP Dimming:1 Service	ISO/IEC 29341-7-10
UPnP SwitchPower:1 Service	ISO/IEC 29341-7-11
UPnP DimmableLight:1 Device	ISO/IEC 29341-7-2
UPnP InternetGatewayDevice:1 Device	ISO/IEC 29341-8-1
UPnP LANHostConfigManagement:1 Service	ISO/IEC 29341-8-10
UPnP Layer3Forwarding:1 Service	ISO/IEC 29341-8-11
UPnP LinkAuthentication:1 Service	ISO/IEC 29341-8-12
UPnP RadiusClient:1 Service	ISO/IEC 29341-8-13
UPnP WANCableLinkConfig:1 Service	ISO/IEC 29341-8-14
UPnP WANCommonInterfaceConfig:1 Service	ISO/IEC 29341-8-15
UPnP WANDSLLinkConfig:1 Service	ISO/IEC 29341-8-16
UPnP WANEthernetLinkConfig:1 Service	ISO/IEC 29341-8-17
UPnP WANIPConnection:1 Service	ISO/IEC 29341-8-18
UPnP WANPOTSLinkConfig:1 Service	ISO/IEC 29341-8-19
UPnP LANDevice:1 Device	ISO/IEC 29341-8-2
UPnP WANPPPConnection:1 Service	ISO/IEC 29341-8-20
UPnP WLANConfiguration:1 Service	ISO/IEC 29341-8-21
UPnP WANDevice:1 Device	ISO/IEC 29341-8-3
UPnP WANConnectionDevice:1 Device	ISO/IEC 29341-8-4
UPnP WLANAccessPointDevice:1 Device	ISO/IEC 29341-8-5
UPnP Printer:1 Device	ISO/IEC 29341-9-1
UPnP ExternalActivity:1 Service	ISO/IEC 29341-9-10
UPnP Feeder:1.0 Service	ISO/IEC 29341-9-11
UPnP PrintBasic:1 Service	ISO/IEC 29341-9-12
UPnP Scan:1 Service	ISO/IEC 29341-9-13
UPnP Scanner:1.0 Device	ISO/IEC 29341-9-2
UPnP QoS Architecture:1.0	ISO/IEC 29341-10-1
UPnP QosDevice:1 Service	ISO/IEC 29341-10-10
UPnP QosManager:1 Service	ISO/IEC 29341-10-11
UPnP QosPolicyHolder:1 Service	ISO/IEC 29341-10-12
UPnP QoS Architecture:2	ISO/IEC 29341-11-1
UPnP QosDevice:2 Service	ISO/IEC 29341-11-10
UPnP QosManager:2 Service	ISO/IEC 29341-11-11
UPnP QosPolicyHolder:2 Service	ISO/IEC 29341-11-12
UPnP QOS v2 Schema Files	ISO/IEC 29341-11-2
UPnP RemoteUIClientDevice:1 Device	ISO/IEC 29341-12-1
UPnP RemoteUIClient:1 Service	ISO/IEC 29341-12-10
UPnP RemoteUIServer:1 Service	ISO/IEC 29341-12-11
UPnP RemoteUIServerDevice:1 Device	ISO/IEC 29341-12-2
UPnP DeviceSecurity:1 Service	ISO/IEC 29341-13-10
UPnP SecurityConsole:1 Service	ISO/IEC 29341-13-11
UPnP ContentDirectory:3 Service	ISO/IEC 29341-14-12:2011
UPnP MediaServer:3 Device	ISO/IEC 29341-14-3:2011

1 Scope

This service definition is compliant with the [1]. It defines a service type referred to herein as *Presence* service.

The *Presence* service is a UPnP service that allows control points, or better known as Telephony Control Points (TelCP), to manage presence information through a Telephony Server (TS) device or a Telephony Client (TC) device (in the role of UPnP server device for *Presence* service). This service provides control points with the following functionality:

- Presence of contacts, access the presence information of contacts, as provided by the UPnP Telephony Server device. Contacts referenced by the *Presence* service may be managed via *PhoneManagement* service and the *Phone Data Model*.

- Managing presence information, update the presence status stored in the TS;

This service does not provide the following functionality:

- To update the presence status of the local UPnP network, from the UPnP server device to remote presence servers (i.e. outside the local UPnP network).

- To retrieve the presence information of contacts from remote presence servers (i.e. outside the local UPnP network).

When the features provided by the *Presence* service are associated to the *PhoneManagement* features, then an integrated management of the address book and the presence of contacts in the address book can be delivered.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

[1] – UPnP Device Architecture, version 1.0, UPnP Forum, October 15, 2008. Available at: http://www.upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.0-20081015.pdf. Latest version available at: http://www.upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.0.pdf.

[2] – Data elements and interchange formats – Information interchange -- Representation of dates and times, International Standards Organization, December 21, 2000. Available at: ISO 8601:2000.

[3] – IETF RFC 2046, Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types, N. Freed, Innosoft, N. Borenstein, First Virtual, November 1996. Available at: http://www.ietf.org/rfc/rfc2046.txt.

[4] – IETF RFC 2119, Key words for use in RFCs to Indicate Requirement Levels, S. Bradner, 1997. Available at: http://www.faqs.org/rfcs/rfc2119.html.

[5] – IETF RFC 2396, Uniform Resource Identifiers (URI): Generic Syntax, T. Berners-Lee, MIT/LCS, R. Fielding, U.C. Irvine, L. Masinter, Xerox Corporation, August 1998. Available at: http://www.ietf.org/rfc/rfc2396.txt

[6] – IETF RFC 3339, Date and Time on the Internet: Timestamps, G. Klyne, Clearswift Corporation, C. Newman, Sun Microsystems, July 2002. Available at: http://www.ietf.org/rfc/rfc3339.txt.

[7] – IETF RFC 3966, The tel URI for Telephone Numbers, H. Schulzrinne, Columbia University, December 2004. Available at: http://www.ietf.org/rfc/rfc3966.txt.

[8] – Extensible Markup Language (XML) 1.0 (Third Edition), François Yergeau, Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, eds., W3C Recommendation, February 4, 2004. Available at: http://www.w3.org/TR/2004/REC-xml-20040204.

[9] – XML Schema Part 2: Data Types, Second Edition, Paul V. Biron, Ashok Malhotra, W3C Recommendation, 28 October 2004. Available at: http://www.w3.org/TR/2004/REC-xmlschema-2-20041028.

[10] – *PhoneManagement:2*, UPnP Forum, December 10, 2012. Available at: http://upnp.org/specs/phone/UPnP-phone-PhoneManagement-v2-Service-20121210.pdf. Latest version available at: http://upnp.org/specs/phone/UPnP-phone-PhoneManagement-Service.pdf.

[11] – IETF RFC 3863, Presence Information Data Format (PIDF), H. Sugano, S. Fujimoto, Fujitsu, G. Klyne, Nine by Nine, A. Bateman, VisionTech, W. Carr, Intel, J. Peterson NeuStar, August 2004. Available at: http://www.ietf.org/rfc/rfc3863.txt.

[12] – IETF RFC 4660, Functional Description of Event Notification Filtering, H. Khartabil, Telio, E. Leppanen, M. Lonnfors, J. Costa-Requena, Nokia, September 2006. Available at: http://www.ietf.org/rfc/rfc4660.txt.

[13] – IETF RFC 4661, An Extensible Markup Language (XML)-Based Format for Event Notification Filtering, H. Khartabil, Telio, E. Leppanen, M. Lonnfors, J. Costa-Requena, Nokia, September 2006. Available at: http://www.ietf.org/rfc/rfc4661.txt.

3   Terms, definitions, symbols and abbreviated terms

For the purposes of this document, the terms and definitions given in [1] and the following apply.

3.1   Provisioning terms

3.1.1

conditionally allowed
CA
The definition or behavior depends on a condition. If the specified condition is met, then the definition or behavior is allowed, otherwise it is not allowed.

3.1.2

conditionally required
CR
The definition or behavior depends on a condition. If the specified condition is met, then the definition or behavior is required, otherwise it is not allowed.

3.1.3

not allowed
The definition or behavior is prohibited by this specification. Opposite of required.

3.2 Symbols

3.2.1

::
signifies a hierarchical parent-child (parent::child) relationship between the two objects separated by the double colon. This delimiter is used in multiple contexts, for example: Service::Action(), Action()::Argument, parentProperty::childProperty.

3.3 General terms

3.3.1
Watcher
Any uniquely identifiable entity that requests the presence information of a TS or user owning the TS.

3.4 Abbreviated terms

3.4.1
GUI
Graphical User Interface

3.4.2
ID
Identifier

3.4.3
TC
Telephony Client

3.4.4
TelCP
Telephony Control Point

3.4.5
TS
Telephony Server

3.4.6
VoIP
Voice over IP

3.4.7
WAN
Wide Area Network

4 Notations and conventions

4.1 Text conventions

- Strings that are to be taken literally are enclosed in "double quotes".

- Words that are emphasized are printed in *italic*.

- Keywords that are defined by the UPnP Working Committee are printed using the *forum* character style.

- Keywords that are defined by [1] are printed using the **arch** character style.

4.1.1 Data Types

This specification uses data type definitions from two different sources. Data types from [1] are used to define state variable and action argument data types [1]. The XML Schema namespace is used to define property data types [9].

For Boolean data types from [1], it is strongly recommended to use the value "**0**" for false, and the value "**1**" for true. The values "**true**", "**yes**", "**false**", or "**no**" may also be used but are not recommended. The values "**yes**" and "**no**" are deprecated and shall not be sent out by devices but shall be accepted on input.

For XML Schema defined Boolean data types, it is strongly recommended to use the value "*0*" for false, and the value "*1*" for true. The values "*true*", "*yes*", "*false*", or "*no*" may also be used but are not recommended. The values "*yes*" and "*no*" are deprecated and shall not be sent out by devices but shall be accepted on input.

4.2 Vendor-defined Extensions

Whenever vendors create additional vendor-defined state variables, actions or properties, their assigned names and XML representation shall follow the naming conventions and XML rules as specified in [1], 2.5, "Description: Non-standard vendor extensions".

5 Service Modeling Definitions

5.1 Service Type

The following service type identifies a service that is compliant with this specification:

   urn:schemas-upnp-org:service:*Presence:1*

Presence service is used herein to refer to this service type..

5.2 *Presence* Service Architecture

This service provides the features for a TelCP to manage the presence information that a UPnP device, either a TS or a TC acting as UPnP server for presence management, exposes in the UPnP network, in order to:

- Retrieve and update the presence status managed by the *Presence* service, and representing the presence information for a user.

- Get notifications of presence updates.

- Retrieve the presence information of the remote contacts managed by the *Presence* service.

- Get notifications of presence updates of remote contacts.

The architecture for the *Presence* service is shown in Figure 1.
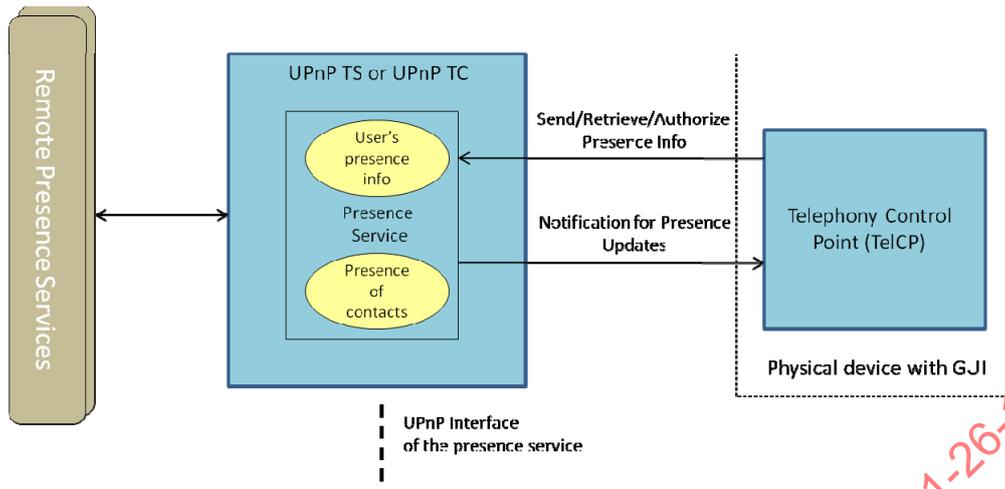
Figure 1 — *Presence* Service Architecture

5.3 State Variables

Note: For first-time reader, it may be more insightful to read the theory of operations first and then the action definitions before reading the state variable definitions.

5.3.1 State Variable Overview

Table 1 — State Variables

Variable Name	R/A a	Data Type	Reference
UserPresenceInfo	*R*	**string** (XML fragment)	See 5.3.2
PresenceOfContactsUpdate	*R*	**string** (XML fragment)	See 5.3.3
A_ARG_TYPE_UserPresenceInfo	*R*	**string** (XML fragment)	See 5.3.4
A_ARG_TYPE_PresenceOfContactsList	*R*	**string** (XML fragment)	See 5.3.5
A_ARG_TYPE_ContactId	*R*	**string** (XML fragment)	See 5.3.6
A_ARG_TYPE_RegistrationResult	*R*	**string**	See 5.3.7
A_ARG_TYPE_Expires	*R*	**ui4**	See 5.3.8
Watcher	*A*	**string** (XML fragment)	See 5.3.9
A_ARG_TYPE_WatcherList	*A*	**string** (XML fragment)	See 5.3.10
a *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*).			

5.3.2 *UserPresenceInfo*

The format of the *UserPresenceInfo* state variable is an XML document. It includes the presence information (e.g., availability of the user) of the user stored in the TS. This state variable is evented to the TelCP(s) in case of any changes to the presence information of a user.

The presence information is represented in Presence Information Data Format (PIDF) format defined in [11].

5.3.2.1 XML Schema Definition

This is a string containing an XML fragment. The XML fragment in this argument shall validate against the XML schema for *UserPresenceInfo* in the XML namespace `"urn:schemas-upnp-org:phone:presence"` which is located at `"http://www.upnp.org/schemas/phone/Presence-v1.xsd"`.

5.3.2.2 Description of fields in the *UserPresenceInfo* structure

```xml
<?xml version="1.0" encoding="UTF-8"?>
<presence:UserPresenceInfo
    xsi:schemaLocation="urn:schemas-upnp-org:phone:presence
    http://www.upnp.org/schemas/phone/presence-v1.xsd"
    xmlns:Presence="urn:schemas-upnp-org:phone:presence"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:pid="urn:ietf:params:xml:ns:pidf">
    <pid:presence entity="pres:someone@example.com">
        <pid:tuple id="bs35r9">
            <pid:status>
                <pid:basic>open</pid:basic>
            </pid:status>
            <pid:contact priority="0.8">someone@mobilecarrier.net</pid:contact>
            <pid:note xml:lang="en">Don't Disturb Please!</pid:note>
            <pid:note xml:lang="fr">Ne derangez pas, s'il vous plait</pid:note>
            <pid:timestamp>2001-10-27T16:49:29Z</pid:timestamp>
        </pid:tuple>
        <!--Any other pid tuple go here -->
    </pid:presence>
</presence:UserPresenceInfo>
```

`<xml>`

Required. Case Sensitive.

`<UserPresenceInfo>`

Required. shall include the name space declaration for the complex type <PIDF> ("urn:ietf:params:xml:ns:pidf") and the namespace declaration for the *Presence* service Schema ("urn:schemas-upnp-org:phone:Presence"). This namespace "urn:schemas-upnp-org:phone:Presence" defines the following elements and attributes:

`<presence>`

Required. includes the presence information in the format defined by [11]. This elemement shall includes "entity" attribute to identify the contact associated with the presence information. This element includes following sub element as defined in the [11]:

`<tuple>`

Allowed. carries a presence information as ordered set of in values. Tuples provide a way of segmenting presence information. The <tuple> element shall contain an 'id' attribute which is used to distinguish this tuple from other tuples. Refer to [11] for more information. This element includes following subelements:

`<status>`

Allowed. represents the availability of a contact. Refer to [11] for more information. This element contains following subelements.

`<basic>`

Allowed. contains one of the following strings: "open" or "closed". This element is used to represent the availability information for the contact.

`<contact>`

Allowed. contains a URL of the contact address. It may have a 'priority' attribute, whose value means a relative priority of this contact address over the others. The value of the attribute shall be a decimal number between 0 and 1. Refer to [11] for more information.

```
<note>
```
> Allowed. contains a string value, which is usually used for a human readable comment.

```
<timestamp>
```
> Allowed. contains a string indicating the date and time of the status change of the presence tuple. Refer to [11] for more information.

This state variable is used to represent the presence information of the user stored in the TS. This specification requires to use the PIDF format to represent the basic presence information (e.g., status information as online/offline) as defined in the [11]. The IETF also have defined extensive rich set of presence information in [15] and [16] as an extension for the basic PIDF format. The *Presence* service can also include any extended presence information into the basic information of this state variable by exporting the particular schema.

5.3.3   *PresenceOfContactsUpdate*

The format of the *PresenceOfContactsUpdate* state variable is an XML document. It includes the presence information of the contact. This state variable is used to notify the TelCP(s) the change in the presence information for the contact. Each time one or more contact presence information is updated, the *Presence* service updates this state variable with the new information only.

The presence information is represented in Presence Information Data Format (PIDF) format defined in [11].

5.3.3.1   XML Schema Definition

This is a string containing an XML fragment. The XML fragment in this argument shall validate against the XML schema for *PresenceOfContactsUpdate* in the XML namespace `"urn:schemas-upnp-org:phone:presence"` which is located at `"http://www.upnp.org/schemas/phone/Presence-v1.xsd"`.

5.3.3.2   Description of fields in the *PresenceOfContactsUpdate* structure

```xml
<?xml version="1.0" encoding="UTF-8"?>
<presence:PresenceOfContactsUpdate
    xsi:schemaLocation="urn:schemas-upnp-org:phone:presence
    http://www.upnp.org/schemas/phone/presence-v1.xsd"
    xmlns:Presence="urn:schemas-upnp-org:phone:presence
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:pid="urn:ietf:params:xml:ns:pidf"
    xmlns:peer="urn:schemas-upnp-org:phone:peer">
    <remotecontact>
        <peer:id>mayur.patil@samsung.com</peer:id>
        <peer:contactInstanceId>12</peer:contactInstanceId>
        <!-- Any Other peer element if required -->
    </remotecontact>
    <pid:presence entity="pres:someone@example.com">
        <pid:tuple id="bs35r9">
            <pid:status>
                <pid:basic>open</pid:basic>
            </pid:status>
            <pid:contact priority="0.8">someone@mobilecarrier.net</pid:contact>
            <pid:note xml:lang="en">Don't Disturb Please!</pid:note>
            <pid:note xml:lang="fr">Ne derangez pas, s'il vous plait</pid:note>
            <pid:timestamp>2001-10-27T16:49:29Z</pid:timestamp>
        </pid:tuple>
        <! -- Any other pid tuple go here -->
    </pid:presence>
</presence:PresenceOfContactsUpdate>
```

`<xml>`

Required. Case Sensitive.

`<PresenceOfContactsUpdate>`

Required. shall include the name space declaration for the complex type <PIDF> ("urn:ietf:params:xml:ns:pidf") and the namespace declaration for the *Presence* service Schema ("urn:schemas-upnp-org:phone:Presence"). This namespace "urn:schemas-upnp-org:phone:Presence" defines the following elements and attributes:

`<remoteContact>`

Required. peer:PeerType, identifies the remote contact whose presence information is updated.

`<presence>`

Required. includes the presence information in the format defined in [11]. This element shall includes "entity" attribute to identify the contact associated with the presence information. This element includes following sub element as defined in [11]:

`<tuple>`

Allowed. carries a presence information as ordered set of in values. Tuples provide a way of segmenting presence information. The <tuple> element shall contain an 'id' attribute which is used to distinguish this tuple from other tuples. Refer to [11] for more information. This element includes following subelements:

`<status>`

Allowed. represents the availability of a contact. Refer to [11] for more information. This element contains following subelements.

`<basic>`

Allowed. contains one of the following strings: "open" or "closed". This element is used to represent the availability information for the contact.

`<contact>`

Allowed. contains a URL of the contact address. It optionally has a 'priority' attribute, whose value means a relative priority of this contact address over the others. The value of the attribute shall be a decimal number between 0 and 1. Refer to [11] for more information.

`<note>`

Allowed. contains a string value, which is usually used for a human readable comment.

`<timestamp>`

Allowed. contains a string indicating the date and time of the status change of the presence tuple. Refer to [11] for more information.

This state variable is used to represent the presence information of the user stored in the TS. This specification requires to use the PIDF format to represent the basic presence information (e.g., status information as online/offline) as defined in the [11]. The IETF also have defined extensive rich set of presence information in [15] and [16] as an extension for the basic PIDF format. The *Presence* service can also include any extended presence information into the basic information of this state variable by exporting the particular schema.

5.3.4 *A_ARG_TYPE_UserPresenceInfo*

The format of the *A_ARG_TYPE_UserPresenceInfo* state variable is an XML document. It includes the presence information of the user stored in the TS.

The presence information is represented in Presence Information Data Format (PIDF) format defined in [11].

5.3.4.1   XML Schema Definition

This is a string containing an XML fragment. The XML fragment in this argument shall validate against the XML schema for *A_ARG_TYPE_UserPresenceInfo* in the XML namespace `"urn:schemas-upnp-org:phone:presence"` which is located at `"http://www.upnp.org/schemas/phone/Presence-v1.xsd"`.

5.3.4.2   Description of fields in the *A_ARG_TYPE_UserPresenceInfo* structure

```
<?xml version="1.0" encoding="UTF-8"?>
<presence:UserPresence
   xsi:schemaLocation="urn:schemas-upnp-org:phone:presence
   http://www.upnp.org/schemas/phone/presence-v1.xsd"
   xmlns:Presence="urn:schemas-upnp-org:phone:presence"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:pid="urn:ietf:params:xml:ns:pidf">
    <pid:presence entity="pres:someone@example.com">
       <pid:tuple id="adgdh">
          <pid:status>
             <pid:basic>open</pid:basic>
          </pid:status>
          <pid:contact priority="0.8">someone@mobilecarrier.net</pid:contact>
          <pid:note xml:lang="en">Don't Disturb Please!</pid:note>
          <pid:note xml:lang="fr">Ne derangez pas, s'il vous plait</pid:note>
          <pid:timestamp>2001-10-27T16:49:29Z</pid:timestamp>
       </pid:tuple>
          <! -- Any other tuple go here -->
    </pid:presence>
</presence:UserPresence>
```

`<xml>`

Required. Case Sensitive.

`<PresenceOfContactsUpdate>`

Required. shall include the namespace declaration for the complex type <PIDF> ("urn:ietf:params:xml:ns:pidf") and the namespace declaration for the *Presence* service Schema ("urn:schemas-upnp-org:phone:Presence"). This namespace "urn:schemas-upnp-org:phone:Presence" defines the following elements and attributes:

`<presence>`

Required. includes the presence information in the format defined in [11]. This elemement shall includes "entity" attribute to identify the contact associated with the presence information. This element includes following sub element as defined in [11]:

`<tuple>`

Allowed. carries a presence information as ordered set of in values. Tuples provide a way of segmenting presence information. The <tuple> element shall contain an 'id' attribute which is used to distinguish this tuple from other tuples. Refer to [11] for more information. This element includes following subelements:

`<status>`

Allowed. represents the availability of a contact. Refer to [11] for more information. This element contains following subelements.

`<basic>`

Allowed. contains one of the following strings: "open" or "closed". This element is used to represent the availability information for the contact.

`<contact>`

Allowed. contains a URL of the contact address. It optionally has a 'priority' attribute, whose value means a relative priority of this contact address over the others. The value of the attribute shall be a decimal number between 0 and 1. Refer to [11] for more information.

```
<note>
```

Allowed. contains a string value, which is usually used for a human readable comment.

```
<timestamp>
```

Allowed. contains a string indicating the date and time of the status change of the presence tuple. Refer to [11] for more information.

This state variable is used to represent the presence information of the user stored in the TS. This specification requires to use the PIDF format to represent the basic presence information (e.g., status information as online/offline) as defined in the [11]. The IETF also have defined extensive rich set of presence information in [15] and [16] as an extension for the basic PIDF format. The *Presence* service can also include any extended presence information into the basic information of this state variable by exporting the particular schema.

5.3.5   *A_ARG_TYPE_PresenceOfContactsList*

The format of the *A_ARG_TYPE_PresenceOfContactsList* state variable is an XML document. It includes the presence information of the contact.

The presence information is represented in Presence Information Data Format (PIDF) format defined in [11].

5.3.5.1   XML Schema Definition

This is a string containing an XML fragment. The XML fragment in this argument shall validate against the XML schema for *A_ARG_TYPE_PresenceOfContactsList* in the XML namespace `"urn:schemas-upnp-org:phone:messaging"` which is located at `"http://www.upnp.org/schemas/phone/Addressbook-v1.xsd"`.

5.3.5.2   Description of fields in the *A_ARG_TYPE_PresenceOfContactsList* structure

```xml
<?xml version="1.0" encoding="UTF-8"?>
<presence:UserPresence
   xsi:schemaLocation="urn:schemas-upnp-org:phone:presence
   http://www.upnp.org/schemas/phone/presence-v1.xsd"
   xmlns:Presence="urn:schemas-upnp-org:phone:presence
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:pid="urn:ietf:params:xml:ns:pidf">
   <pid:presence entity="pres:someone@example.com">
      <pid:tuple id="adgdh">
         <pid:status>
            <pid:basic>open</pid:basic>
         </pid:status>
         <pid:contact priority="0.8">someone@mobilecarrier.net</pid:contact>
         <pid:note xml:lang="en">Don't Disturb Please!</pid:note>
         <pid:note xml:lang="fr">Ne derangez pas, s'il vous plait</pid:note>
         <pid:timestamp>2001-10-27T16:49:29Z</pid:timestamp>
      </pid:tuple>
      <! -- Any other tuple go here -->
   </pid:presence>
</presence:UserPresence>
```

```
<xml>
```

Required. Case Sensitive.

```
<PresenceOfContactsUpdate>
```

Required. shall include the name space declaration for the complex type <PIDF> ("urn:ietf:params:xml:ns:pidf") and the namespace declaration for the *Presence* service Schema ("urn:schemas-upnp-org:phone:Presence"). This namespace "urn:schemas-upnp-org:phone:Presence" defines the following elements and attributes:

```
<presence>
```

Required. includes the presence information in the format defined in [11]. This elemement shall includes "entity" attribute to identify the contact associated with the presence information. This element includes following sub element as defined in [11]:

```
<tuple>
```

Allowed. carries a presence information as ordered set of in values. Tuples provide a way of segmenting presence information. The <tuple> element shall contain an 'id' attribute which is used to distinguish this tuple from other tuples. Refer to [11] for more information. This element includes following subelements:

```
<status>
```

Allowed. represents the availability of a contact. Refer to [11] for more information. This element contains following subelements.

```
<basic>
```

Allowed. contains one of the following strings: "open" or "closed". This element is used to represent the availability information for the contact.

```
<contact>
```

Allowed. contains a URL of the contact address. It optionally has a 'priority' attribute, whose value means a relative   priority of this contact address over the others. The value of the attribute shall be a decimal number between 0 and 1. Refer to [11] for more information.

```
<note>
```

Allowed. contains a string value, which is usually used for a human readable comment.

```
<timestamp>
```

Allowed. contains a string indicating the date and time of the status change of the presence tuple. Refer to [11] for more information.

This state variable is used to represent the presence information of the user stored in the TS. This specification requires to use the PIDF format to represent the basic presence information (e.g., status information as online/offline) as defined in the [11]. The IETF also have defined extensive rich set of presence information in [15] and [16] as an extension for the basic PIDF format. The *Presence* service can also include any extended presence information into the basic information of this state variable by exporting the particular schema.

5.3.6   *A_ARG_TYPE_ContactId*

This state variable identifies one or more contact(s). This element is an XML structure of type *Peer*.

5.3.6.1   Description of fields in the *A_ARG_TYPE_ContactId*

This is a string containing an XML fragment. The XML fragment in this argument shall validate against the XML schema for *A_ARG_TYPE_ContactId* in the XML namespace "`urn:schemas-upnp-org:phone:presence`" which is located at "`http://www.upnp.org/schemas/phone/Presence-v1.xsd`".

```
<?xml version="1.0" encoding="UTF-8"?>
<contactId
   xsi:schemaLocation="urn:schemas-upnp-org:phone:presence
   http://www.upnp.org/schemas/phone/presence-v1.xsd"
   xmlns:Presence="urn:schemas-upnp-org:phone:presence
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:peer="urn:schemas-upnp-org:phone:peer">
   <contact>
      <peer:id>ID of the peer</peer:id>
      <peer:contactInstanceId>instance id of the contact</peer:contactInstanceId>)
```

```
        Other peer elements if required…
    </contact>
        <!-- Any other contact information (if any) go here.-->
</contactId>
```

`<xml>`

> Required. Case Sensitive.

`<contactId>`

> Required. shall include the name space declaration for the complex type <peerType> ("urn:schemas-upnp-org:phone:peer") and the namespace declaration for the *Presence* service Schema ("urn:schemas-upnp-org:phone:presence"). This namespace "urn:schemas-upnp-org:phone:presence" defines the following elements and attributes:

> `<contact>`

> > Required, peer:peerType. identifies the contact to be shared. If contact to be shared is present in the local address book then,the contactInstanceId element may be included to refer the contact from the address book otherwise the Id element will identify the contact. A TelCP can use the any other peer elements (e.g., name etc) to identify the contact.

5.3.7   *A_ARG_TYPE_RegistrationResult*

This state variable contains the subscription request result. This state variable is of a type string. This state variable contains one of the following values.

- "accepted".
- "rejected".
- "pending".

5.3.8   *A_ARG_TYPE_Expires*

This state variable contains the duration as an integer value.

5.3.9   *Watcher*

The format of the *Watcher* state variable is an XML document. It includes the remote party contact information who is requesting for the presence information. This state variable is evented to the TelCP(s), when TS receives an incoming request from the remote party for sharing the presence information.

5.3.9.1   Description of fields in the *Watcher*

This is a string containing an XML fragment. The XML fragment in this argument shall validate against the XML schema for *Watcher* in the XML namespace "`urn:schemas-upnp-org:phone:presence`" which is located at "`http://www.upnp.org/schemas/phone/Presence-v1.xsd`".

```
<?xml version="1.0" encoding="UTF-8"?>
<presence:Watcher
  xsi:schemaLocation="urn:schemas-upnp-org:phone:presence
  http://www.upnp.org/schemas/phone/presence-v1.xsd"
  xmlns:Presence="urn:schemas-upnp-org:phone:presence
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:peer="urn:schemas-upnp-org:phone:peer"
  xmlns:pid="urn:ietf:params:xml:ns:pidf">
  <watcher_info>
     <peer:id>ID of the peer</peer:id>
     <peer:contactInstanceId>instance ID of the contact</peer:contactInstanceId>
  </watcher_info>
  <presence_info format= "pidf">
     <-- Presence information requested by the watcher represented in pidf -->
  </presence_info>
```

```
</presence:Watcher>
```

`<xml>`

Required. Case Sensitive.

`<Watcher>`

Required. shall include the name space declaration for the complex type <PeerType> ("urn:schemas-upnp-org:phone:peer") and the namespace declaration for the *Presence* service Schema ("urn:schemas-upnp-org:phone:Presence"). This namespace "urn:schemas-upnp-org:phone:Presence" defines the following elements and attributes:

`<watcher_info>`

Required. peer:PeerType, identifies the remote contact requesting the presence information.

`<peer:id>`

Allowed. identifies the remote contact who is requesting for the presence information.

`<peer:contactInstanceId>`

Allowed. references the contact from the address book who is requesting for the presence information.

`<presence_info>`

Allowed. represents the requested presence information (or filters) by the remote party. The format attribute will specify the format in which the presence information is specified. The TS includes the presence information of the user which is requested by the watcher in the PIDF format.

5.3.10 *A_ARG_TYPE_WatcherList*

The format of the *A_ARG_TYPE_WatcherList* state variable is an XML document. It includes the list of Watchers currently requesting or already subscribed for the presence information.

5.3.10.1 Description of fields in the *A_ARG_TYPE_WatcherList*

This is a string containing an XML fragment. The XML fragment in this argument shall validate against the XML schema for *A_ARG_TYPE_WatcherList* in the XML namespace `"urn:schemas-upnp-org:phone:presence"` which is located at `"http://www.upnp.org/schemas/phone/Presence-v1.xsd"`.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<presence:WatcherList
    xsi:schemaLocation="urn:schemas-upnp-org:phone:presence
    http://www.upnp.org/schemas/phone/presence-v1.xsd"
    xmlns:Presence="urn:schemas-upnp-org:phone:presence
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:peer="urn:schemas-upnp-org:phone:peer">
    <watcher>
     <watcher_info>
        <peer:id>ID of the peer</peer:id>
        <peer:contactInstanceId>instanceID of the contact</peer:contactInstanceId>
    </watcher_info>
    <presence_info format="RFC4660/RFC4661/URL">
        requested presence information
    </presence_info>
    </watcher>
    <!-- Any other watcher information goes here -->
</presence:WatcherList>
```

`<xml>`

Required. Case Sensitive.

```
<WatcherList>
```

> Required. shall include the name space declaration for the complex type <PeerType> ("urn:schemas-upnp-org:phone:peer") and the namespace declaration for the *Presence* service Schema ("urn:schemas-upnp-org:phone:Presence"). This namespace "urn:schemas-upnp-org:phone:Presence" defines the following elements and attributes:

```
<watcher>
```

> Required. carries the Watcher information and current status of the subscription. This elements has following sub-elements.

```
<watcher_info>
```

> Required. peer:PeerType, identifies the remote contact requesting the presence information.

```
<peer:id>
```

> Allowed. identifies the remote contact who is requesting for the presence information.

```
<peer:contactInstanceId>
```

> Allowed. references the contact from the address book who is requesting for the presence information.

```
<presence_info>
```

> Allowed. represents the requested presence information (or filters) by the remote party. The format attribute will specify the format in which the filters are specified. The IETF has defined the filtering mechanism in [12], [13]. The TS export the structure defined by the corresponding RFCs.

5.4 Eventing and Moderation

Table 2 — Event Moderation

Variable Name	Evented	Moderated Event	Max Event Rate [a] (seconds)	Logical Combination	Min Delta per Event [b]
UserPresenceInfo	*YES*	*YES*	1		
PresenceOfContactsUpdate	*YES*	*YES*	1		
Watcher	*YES*	*YES*	1		
A_ARG_TYPE_UserPresenceInfo	*No*	*No*			
A_ARG_TYPE_PresenceOfContactsList	*No*	*No*			
A_ARG_TYPE_ContactId	*No*	*No*			
A_ARG_TYPE_RegistrationResult	*No*	*No*			
A_ARG_TYPE_Expires	*No*	*No*			
A_ARG_TYPE_WatcherList	*No*	*No*			
[a] Determined by N, where Rate = (Event)/(N secs).					
[b] (N) * (allowedValueRange Step).					

5.4.1 Eventing of *UserPresenceInfo*

This state variable is evented when a user presence information changes. However, this state variable shall not be evented more than once in 1 s. If multiple changes arrive within 1 s, then all the events will be accumulated into a single event message and the event message will be sent after 1 s.

5.4.2 Eventing of *PresenceOfContactsUpdate*

This state variable is evented when presence information of a particular contact changes. However, this state variable shall not be evented more than once in 1 s. If multiple change requests arrive within 1 s, then all the events will be accumulated into a single event message and the event message will be sent after 1 s.

5.4.3 Eventing of *Watcher*

This state variable is evented when a TS receives the request from the contact for sharing the presence information from WAN side. However, this state variable shall not be evented more than once in 1 s. If multiple requests arrive within 1 s, then all the events will be accumulated into a single event message and the event message will be sent after 1 s.

5.5 Actions

Table 3 lists the actions of the *Presence* service.

The *Presence* service requires the implementation of all the actions necessary for delivering the complete set of functions.

Table 3 — Actions

Name	Device R/A [a]	Control Point R/A [b]
GetPresence()	*A*	*A*
UpdatePresence()	*R*	*R*
GetContactPresence()	*R*	*R*
GetPresenceOfContactsUpdate()	*A*	*A*
RegisterForContactPresence()	*R*	*R*
AuthorizePresenceReactive()	*A*	*A*
AuthorizePresenceProactive()	*A*	*A*

[a] For a device this column indicates whether the action shall be implemented or not, where *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*).

[b] For a control pont this column indicates whether a control point shall be capable of invoking this action, where *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*).

5.5.1 *GetPresence()*

This action allows a TelCP to retrieve the user's presence information stored in the TS.

5.5.1.1 Arguments

Table 4 — Arguments for *GetPresence()*

Argument	Direction	relatedStateVariable
UserPresence	*OUT*	*A_ARG_TYPE_UserPresenceInfo*

5.5.1.2 Argument Descriptions

The output argument *UserPresence* returns the user's current presence information.

5.5.1.3   Service Requirements

None.

5.5.1.4   Control Point Requirements When Calling The Action

None.

5.5.1.5   Dependency on Device State

None.

5.5.1.6   Effect on Device State

None.

5.5.1.7   Errors

Table 5 — Error Codes for *GetPresence()*

ErrorCode	errorDescription	Description
400-499	TBD	See Control clause in [1].
500-599	TBD	See Control clause in [1].
606	Action not Authorized	The CP does not have privileges to invoke this action.

5.5.2   *UpdatePresence()*

This action allows a TelCP to updates the user's presence information in the TS.

5.5.2.1 Arguments

Table 6 — Arguments for *UpdatePresence()*

Argument	Direction	relatedStateVariable
UpdatedUserPresence	*IN*	*A_ARG_TYPE_UserPresenceInfo*

5.5.2.2 Argument Descriptions

The input argument *UpdatedUserPresence* will include the updated presence information.

5.5.2.3 Service Requirements

The input argument will carry the updated presence information. Only the tuple elements included in the input argument will be updated in the presence information of the user, other tuple elements in the presence information will not be modified.

5.5.2.4 Control Point Requirements When Calling The Action

None.

5.5.2.5 Dependency on Device State

None.

5.5.2.6 Effect on Device State

None.

5.5.2.7 Errors

Table 7 — Error Codes for *UpdatePresence()*

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture clause on Control.
500-599	TBD	See UPnP Device Architecture clause on Control.
600-699	TBD	See UPnP Device Architecture clause on Control.

5.5.3 *GetContactPresence()*

This action allows a TelCP to retrieve the presence information of a contact.

5.5.3.1 Arguments

Table 8 — Arguments for *GetContactPresence()*

Argument	Direction	relatedStateVariable
TargetContact	*IN*	*A_ARG_TYPE_ContactId*
ContactPresence	*OUT*	*A_ARG_TYPE_PresenceOfContactsList*

5.5.3.2 Argument Descriptions

The input argument *TargetContact* identifies the target contact.

The output argument *ContactPresence* contains the presence information of the contact.

5.5.3.3 Service Requirements

None.

5.5.3.4 Control Point Requirements When Calling The Action

None.

5.5.3.5 Dependency on Device State

None.

5.5.3.6 Effect on Device State

None.

5.5.3.7 Errors

Table 9 — Error Codes for *GetContactPresence()*

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture clause on Control.
500-599	TBD	See UPnP Device Architecture clause on Control.
600-699	TBD	See UPnP Device Architecture clause on Control.
770	Invalid target contact	The target contact does not exist.

5.5.4 *GetPresenceOfContactsUpdate()*

This action allows a TelCP to retrieve the updated presence information for the contact(s) which was evented by the *PresenceOfContactsUpdate* state variable.

5.5.4.1 Arguments

Table 10 — Arguments for *GetPresenceOfContactsUpdate()*

Argument	Direction	relatedStateVariable
ContactPresenceUpdate	*OUT*	*A_ARG_TYPE_PresenceOfContactsList*

5.5.4.2 Argument Descriptions

The output argument *ContactPresenceUpdate* contains the presence information of the contact(s).

5.5.4.3 Service Requirements

None.

5.5.4.4 Control Point Requirements When Calling The Action

None.

5.5.4.5 Dependency on Device State

None.

5.5.4.6 Effect on Device State

None.

5.5.4.7 Errors

Table 11 — Error Codes for *GetPresenceOfContactsUpdate()*

ErrorCode	errorDescription	Description

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture clause on Control.
500-599	TBD	See UPnP Device Architecture clause on Control.
600-699	TBD	See UPnP Device Architecture clause on Control.

5.5.5 *RegisterForContactPresence()*

This action allows a TelCP to subscribe for the presence information of a contact. The same action can be used to unsubscribe the presence information for a contact by setting the value of the input argument *Expire* as"0".

5.5.5.1 Arguments

Table 12 — Arguments for *RegisterForContactPresence()*

Argument	Direction	relatedStateVariable
Contact	*IN*	*A_ARG_TYPE_ContactId*
Expire	*IN*	*A_ARG_TYPE_Expires*
RegistrationResult	*OUT*	*A_ARG_TYPE_RegistrationResult*

5.5.5.2 Argument Descriptions

The input argument *Contact* identifies the remote party.

The input argument *Expire* contains the duration for the subscription. If the value of the *Expire* input argument is set to "0" then the current subscription will be terminated by the TS.

The output argument *RegistrationResult* is used to notify the result for the subscription request.

5.5.5.3 Service Requirements

None.

5.5.5.4 Control Point Requirements When Calling The Action

None.

5.5.5.5 Dependency on Device State

None.

5.5.5.6 Effect on Device State

None.

5.5.5.7 Errors

Table 13 — Error Codes for *RegisterForContactPresence()*

ErrorCode	errorDescription	Description

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture clause on Control.
500-599	TBD	See UPnP Device Architecture clause on Control.
600-699	TBD	See UPnP Device Architecture clause on Control.
770	Invalid target contact	The target contact does not exist.

5.5.6 *AuthorizePresenceReactive()*

This action allows a TelCP to authorize a contact to view the presence information stored in the Telephony Server (TS). This action is invoked in response to an event sent by the TS. When the TS receives a request for the presence information from the WAN side, the TS sends out event notifications to all the subscribed control points in the home network about the incoming requests. A TelCP can then accept or reject the request by invoking this action on the TS, which is then propagated to the requester using a WAN-side protocol by the presence server.

5.5.6.1 Arguments

Table 14 — Arguments for *AuthorizePresenceReactive()*

Argument	Direction	relatedStateVariable
Contact	*IN*	*A_ARG_TYPE_ContactId*
Expire	*IN*	*A_ARG_TYPE_Expires*
UserPresenceInfo	*IN*	*A_ARG_TYPE_UserPresenceInfo*

5.5.6.2 Argument Descriptions

The input argument *Contact* identifies the remote peer contact that requested for the presence information of the user.

The input argument *Expire* contains the duration for which this authorization is valid. If the value of the input argument *Expire* is set to "0", then the authorization to view the presence information of the user is denied.

The input argument *UserPresenceInfo* specifies the presence information to be shared with the contact.

5.5.6.3 Service Requirements

None.

5.5.6.4 Control Point Requirements When Calling The Action

None.

5.5.6.5 Dependency on Device State

None.

5.5.6.6 Effect on Device State

None.

5.5.6.7 Errors

Table 15 — Error Codes for *AuthorizePresenceReactive()*

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture clause on Control.
500-599	TBD	See UPnP Device Architecture clause on Control.
600-699	TBD	See UPnP Device Architecture clause on Control.
770	Invalid contact	The contact does not exist

5.5.7 *AuthorizePresenceProactive()*

This action allows a TelCP to authorize (allow or block) a list of contacts to view the presence information of the user stored in the TS. The list of entities on the WAN side that are authorized to view the presence information of the user (TS with Presence Service) can be set up by this action even before the request for presence information is received by the TS. The TelCP invokes this action with a list of WAN contacts that are allowed to access the presence information.

5.5.7.1 Arguments

Table 16 — Arguments for *AuthorizePresenceProactive()*

Argument	Direction	relatedStateVariable
UserPresenceInfo	*IN*	*A_ARG_TYPE_UserPresenceInfo*
Expire	*IN*	*A_ARG_TYPE_Expires*
WatcherList	*IN*	*A_ARG_TYPE_WatcherList*

5.5.7.2 Argument Descriptions

The input argument *UserPresenceInfo* contains the presence information of the user which a watcher is authorized to view.

The input argument *Expire* contains the duration for which this authorization is valid. If the value of the input argument *Expire* is set to "0", then the authorization to view the presence information is blocked for the contacts in the *WatcherList*.

The output argument *WatcherList* includes the list of contacts that can view the presence information of the user.

5.5.7.3 Service Requirements

When the presence service within a TS receives a request to access presence information from a contact, the TS checks its internal database to see whether the contact is allowed to access the presence information. The server can then decide to allow or deny the request to the WAN side without even consulting the devices in the home network. This is proactive authorization mechanism.

5.5.7.4 Control Point Requirements When Calling The Action

None.

5.5.7.5 Dependency on Device State

None.

5.5.7.6 Effect on Device State

None.

5.5.7.7 Errors

Table 17 — Error Codes for _AuthorizePresenceProactive()_

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture clause on Control.
500-599	TBD	See UPnP Device Architecture clause on Control.
600-699	TBD	See UPnP Device Architecture clause on Control.

5.5.8   Error Code Summary

Table 18 lists error codes common to actions for this service type. If an action results in multiple errors, the most specific error should be returned.

Table 18 — Error Code Summary

ErrorCode	errorDescription	Description
400-499	TBD	See Control clause in [1].
500-599	TBD	See Control clause in [1].
606	Action not Autohrized	The CP does not have privileges to invoke this action.
770	Invalid contact	The contact does not exist.

Note: 800-899 Error Codes are not permitted for standard actions. See Control clause in [1] for more details.

5.6   Service Behavioral Model

None.

6   XML Service Description

```
<?xml version="1.0"?>
<scpd xmlns="urn:schemas-upnp-org:service-1-0">

  <specVersion>
      <major>1</major>
      <minor>0</minor>
  </specVersion>

  <actionList>

    <action>
        <name>GetPresence</name>
        <argumentList>
          <argument>
            <name>UserPresence</name>
            <direction>OUT</direction>
```

```xml
            <relatedStateVariable>
                A_ARG_TYPE_UserPresenceInfo
            </relatedStateVariable>
        </argument>
    </argumentList>
</action>

<action>
    <name>UpdatePresence</name>
    <argumentList>
        <argument>
            <name>UpdatedUserPresence</name>
            <direction>in</direction>
            <relatedStateVariable>
                A_ARG_TYPE_UserPresenceInfo
            </relatedStateVariable>
        </argument>
    </argumentList>
</action>

<action>
    <name>GetContactPresence</name>
    <argumentList>
        <argument>
            <name>TargetContact</name>
            <direction>in</direction>
            <relatedStateVariable>
                A_ARG_TYPE_ContactId
            </relatedStateVariable>
        </argument>

        <argument>
            <name>ContactPresence</name>
            <direction>out</direction>
            <relatedStateVariable>
                A_ARG_TYPE_PresenceOfContactsList
            </relatedStateVariable>
        </argument>
    </argumentList>
</action>

<action>
    <name>GetPresenceOfContactsUpdate</name>
    <argumentList>
        <argument>
            <name>ContactPresenceUpdate</name>
            <direction>out</direction>
            <relatedStateVariable>
                A_ARG_TYPE_PresenceOfContactsList
            </relatedStateVariable>
        </argument>
    </argumentList>
</action>

<action>
    <name>RegisterForContactPresence</name>
    <argumentList>
        <argument>
            <name>Contact</name>
            <direction>in</direction>
            <relatedStateVariable>
                A_ARG_TYPE_ContactId
            </relatedStateVariable>
        </argument>

        <argument>
            <name>Expire</name>
            <direction>in</direction>
```

```
            <relatedStateVariable>
               A_ARG_TYPE_Expires
            </relatedStateVariable>
         </argument>

         <argument>
            <name>RegistrationResult</name>
            <direction>out</direction>
            <relatedStateVariable>
               A_ARG_TYPE_RegistrationResult
            </relatedStateVariable>
         </argument>
      </argumentList>
   </action>

   <action>
      <name>AuthorizePresenceReactive</name>
      <argumentList>
         <argument>
            <name>Contact</name>
            <direction>in</direction>
            <relatedStateVariable>
            A_ARG_TYPE_ContactId
            </relatedStateVariable>
         </argument>

         <argument>
            <name>Expire</name>
            <direction>in</direction>
            <relatedStateVariable>
               A_ARG_TYPE_Expires
            </relatedStateVariable>
         </argument>

         <argument>
            <name>UserPresenceInfo</name>
            <direction>in</direction>
            <relatedStateVariable>
               A_ARG_TYPE_UserPresenceInfo
            </relatedStateVariable>
         </argument>
      </argumentList>
   </action>

   <action>
      <name> AuthorizePresenceProactive</name>
      <argumentList>
         <argument>
            <name>UserPresenceInfo</name>
            <direction>in</direction>
            <relatedStateVariable>
               A_ARG_TYPE_UserPresenceInfo
            </relatedStateVariable>
         </argument>

         <argument>
            <name>Expire</name>
            <direction>in</direction>
            <relatedStateVariable>
               A_ARG_TYPE_Expires
            </relatedStateVariable>
         </argument>

         <argument>
            <name>WatcherList</name>
            <direction>in</direction>
            <relatedStateVariable>
               A_ARG_TYPE_WatcherList
```

```xml
                        </relatedStateVariable>
                    </argument>
                </argumentList>
            </action>

    </actionList>

    <serviceStateTable>

        <stateVariable sendEvents="yes">
            <name>UserPresenceInfo</name>
            <dataType>string</dataType>
        </stateVariable>

        <stateVariable sendEvents="yes">
            <name>PresenceOfContactsUpdate</name>
            <dataType>string</dataType>
        </stateVariable>

        <stateVariable sendEvents="yes">
            <name>Watcher</name>
            <dataType>string</dataType>
        </stateVariable>


        <stateVariable sendEvents="no">
            <name>A_ARG_TYPE_WatcherList</name>
            <dataType>string</dataType>
        </stateVariable>

        <stateVariable sendEvents="no">
            <name>A_ARG_TYPE_UserPresenceInfo</name>
            <dataType>string</dataType>
        </stateVariable>


        <stateVariable sendEvents="no">
            <name>A_ARG_TYPE_PresenceOfContactsList</name>
            <dataType>string</dataType>
        </stateVariable>

        <stateVariable sendEvents="no">
            <name>A_ARG_TYPE_ContactId</name>
            <dataType>string</dataType>
        </stateVariable>

        <stateVariable sendEvents="no">
            <name>A_ARG_TYPE_RegistrationResult</name>
            <dataType>string</dataType>
        </stateVariable>
        <stateVariable sendEvents="no">
            <name>A_ARG_TYPE_Expires</name>
            <dataType>Integer</dataType>
        </stateVariable>

    </serviceStateTable>
</scpd>
```

Annex A
(normative)

XML complex type *peerType*

A communication means the exchange of an information between two or more end entities. These end entities are herein referred as Peers. The Peer can be a caller of a phone call, recipient of an email message, or group of participants in a communication session, or a contact in an Address book.

In order to have a uniform representation of a Peer across all the services in the UPnP Telephony, the XML complex type peerType is defined. The same XML complex type can be reused by other UPnP Telephony services.

The complex type peerType contains the information to properly identify a contact and its communication address for e.g. a phone call needs a telephone number, an email message needs an email address etc. Along with the communication address it is also important to include additional information about the Peer for e.g. photo, location information of user etc. If TS supports the PhoneManagement profile, then the correspondence between the Peer element and either a contact or a group of contacts in the Address book is also included in the complex peerType element.

A.1   Using the *peerType* within XML Schemas

The complex type `peerType` can be used in the XML schemas by including the following statement:

```
<import
    namespace="urn:schemas-upnp-org:phone:peer"
    schemaLocation="http://www.upnp.org/schemas/phone/peer-v2.xsd"/>
```

where the `schemaLocation` refers to the last updated schema file for the *Peer.*

A.2   Description of fields of a `peerType` complex type

Clause A.2 gives a description of the elements defined in the peerType complex type.

```
<?xml version="1.0" encoding="UTF-8"?>
<peer:peer
    xsi:schemaLocation="urn:schemas-upnp-org:phone:peer
    http://www.upnp.org/schemas/phone/peer-v2.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:peer="urn:schemas-upnp-org:phone:peer">
    <peer:id>
        The identifier of the Peer (e.g., phone number, etc.)
    </peer:id>
    <peer:name>A user friendly name for the Peer</peer:name>
    <peer:contactInstanceId>
        The instance identifier for a contact referred by the Peer
    </peer:contactInstanceId>
    <peer:groupInstanceId>
        The instance identifier for a group referred by the Peer
    </peer:groupInstanceId>
    <peer:image type="URL"/"thumbnail">
        <peer:contentType>
            content type information for the image data
        </peer:contentType>
        <peer:contentTransferEncoding>
            Encoding information
```