INTERNATIONAL
STANDARD

ISO/IEC
29341-26-13

First edition
2017-09

Information technology — UPnP Device Architecture —

Part 26-13:
Telephony device control protocol — Level 2 — Phone management service

Technologies de l'information — Architecture de dispositif UPnP —

Partie 26-13: Protocole de contrôle de dispositif de téléphonie — Niveau 2 — Service de gestion du téléphone

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 29341-26-13 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

UPnP ContentSync:1	ISO/IEC 29341-15-10:2011
UPnP Low Power Architecture:1	ISO/IEC 29341-16-1:2011
UPnP LowPowerProxy:1 Service	ISO/IEC 29341-16-10:2011
UPnP LowPowerDevice:1 Service	ISO/IEC 29341-16-11:2011
UPnP QoS Architecture:3	ISO/IEC 29341-17-1:2011
UPnP QosDevice:3 Service	ISO/IEC 29341-17-10:2011
UPnP QosManager:3 Service	ISO/IEC 29341-17-11:2011
UPnP QosPolicyHolder:3 Service	ISO/IEC 29341-17-12:2011
UPnP QosDevice:3 Addendum	ISO/IEC 29341-17-13:2011
UPnP RemoteAccessArchitecture:1	ISO/IEC 29341-18-1:2011
UPnP InboundConnectionConfig:1 Service	ISO/IEC 29341-18-10:2011
UPnP RADAConfig:1 Service	ISO/IEC 29341-18-11:2011
UPnP RADASync:1 Service	ISO/IEC 29341-18-12:2011
UPnP RATAConfig:1 Service	ISO/IEC 29341-18-13:2011
UPnP RAClient:1 Device	ISO/IEC 29341-18-2:2011
UPnP RAServer:1 Device	ISO/IEC 29341-18-3:2011
UPnP RADiscoveryAgent:1 Device	ISO/IEC 29341-18-4:2011
UPnP SolarProtectionBlind:1 Device	ISO/IEC 29341-19-1:2011
UPnP TwoWayMotionMotor:1 Service	ISO/IEC 29341-19-10:2011
UPnP AV Architecture:2	ISO/IEC 29341-20-1
UPnP AVTransport:3 Service	ISO/IEC 29341-20-10
UPnP ConnectionManager:3 Service	ISO/IEC 29341-20-11
UPnP ContentDirectory:4 Device	ISO/IEC 29341-20-12
UPnP RenderingControl:3 Service	ISO/IEC 29341-20-13
UPnP ScheduledRecording:2 Service	ISO/IEC 29341-20-14
UPnP MediaRenderer:3 Service	ISO/IEC 29341-20-2
UPnP MediaServer:4 Device	ISO/IEC 29341-20-3
UPnP AV Datastructure Template:1	ISO/IEC 29341-20-4
UPnP InternetGatewayDevice:2 Device	ISO/IEC 29341-24-1
UPnP WANIPConnection:2 Service	ISO/IEC 29341-24-10
UPnP WANIPv6FirewallControl:1 Service	ISO/IEC 29341-24-11
UPnP WANConnectionDevice:2 Service	ISO/IEC 29341-24-2
UPnP WANDevice:2 Device	ISO/IEC 29341-24-3
UPnP Telephony Architecture:2	ISO/IEC 29341-26-1
UPnP CallManagement:2 Service	ISO/IEC 29341-26-10
UPnP MediaManagement:2 Service	ISO/IEC 29341-26-11
UPnP Messaging:2 Service	ISO/IEC 29341-26-12
UPnP PhoneManagement:2 Service	ISO/IEC 29341-26-13
UPnP AddressBook:1 Service	ISO/IEC 29341-26-14
UPnP Calendar:1 Service	ISO/IEC 29341-26-15
UPnP Presense:1 Service	ISO/IEC 29341-26-16
UPnP TelephonyClient:2 Device	ISO/IEC 29341-26-2
UPnP TelephonyServer:2 Device	ISO/IEC 29341-26-3
UPnP Friendly Info Update:1 Service	ISO/IEC 29341-27-1
UPnP MultiScreen MultiScreen Architecture:1	ISO/IEC 29341-28-1
UPnP MultiScreen Application Management:1 Service	ISO/IEC 29341-28-10

UPnP MultiScreen Screen:1 Device	ISO/IEC 29341-28-2
UPnP MultiScreen Application Management:2 Service	ISO/IEC 29341-29-10
UPnP MultiScreen Screen:2 Device	ISO/IEC 29341-29-2
UPnP IoT Management and Control Architecture Overview:1	ISO/IEC 29341-30-1
UPnP DataStore:1 Service	ISO/IEC 29341-30-10
UPnP IoT Management and Control Data Model:1 Service	ISO/IEC 29341-30-11
UPnP IoT Management and Control Transport Generic:1 Service	ISO/IEC 29341-30-12
UPnP IoT Management and Control:1 Device	ISO/IEC 29341-30-2
UPnP Energy Management:1 Service	ISO/IEC 29341-31-1

1 Scope

This document specifies the *PhoneManagement* profile of the *ConfigurationManagement* Service defined in [7].

The *PhoneManagement* profile can be used for managing the configuration of the UPnP *TelephonyServer* device (i.e. a telephone), with tasks such as managing an address book, configuring the settings of the phone, configuring the ringing modes, checking the battery level of the phone. This profile is compliant with the UPnP Telephony Architecture defined in [9].

PhoneManagement defines

* A profile of the CMS in order to satisfy the requirements for *PhoneManagement* (see clause 5).

* The *Phone Data Model* to perform configuration management of the specific features of a phone (see Annex A) by reusing the CMS data model (see [7], Annex B).

The *ConfigurationManagement* service offers a general purpose data model and a set of configuration management operations for retrieving and managing the actual configuration parameters of a device. These features are completely reused from CMS in the *PhoneManagement*.

The *ConfigurationManagement* service defines the functions for manipulating the configuration and status parameters that are exposed by the device hosting the *ConfigurationManagement* service. These actions and state variables defined in the *ConfigurationManagement* service are reused in order to manage the configuration parameters of a *TelephonyServer* device for e.g. phone, defined in the data model of the *PhoneManagement*.

The main goal of this specification is to define the data model (called herein as *Phone Data Model*) for the *TelephonyServer* device like Phone, according to the rules defined by the *ConfigurationManagement* service for defining a new data model.

The *Phone Data Model* is organized as a hierarchical tree of parameter sets, where each set represent a feature of a *TelephonyServer* (e.g. address book, ringing modes etc) that can be managed by a TelCP.

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

[4] – 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.

[5] – 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.

[6] – 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.

[7] – *ConfigurationManagement:1 Service*, UPnP Forum, July 5, 2010. Available at: http://www.upnp.org/specs/dm/UPnP-dm-ConfigurationManagement-v1-Service.pdf. Latest version available at: http://www.upnp.org/specs/dm/UPnP-dm-ConfigurationManagement-Service.pdf.

[8] – IETF RFC 2426, vCard MIME Directory Profile, F. Dawson, Lotus Development Corporation, T. Howes, Netscape Communications, September 1998. Available at: http://www.ietf.org/rfc/rfc2426.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 Abbreviated terms

3.3.1

A
allowed

3.3.2

CSV list
CSV
comma separated value list
list—or one-dimensional array—of values contained in a string and separated by commas

3.3.3

CMS
Configuration Management Service

3.3.4

TC
Telephony Client

3.3.5

TelCP
Telephony Control Point

3.3.6

TS
Telephony Server

4   Notations and conventions

4.1   Notation

- 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. The UPnP Device Architecture defined data types are used to define state variable and action argument data types [1]. The XML Schema namespace is used to define property data types [6].

For UPnP Device Architecture 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.

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.

5 *ConfigurationManagement* Service Profile

The *PhoneManagement* is a **profile** of the *ConfigurationManagement* service [7] which specifies:

- The requirements on the attributes and the related actions of the *ConfigurationManagement* service.
- The required state variables for the *PhoneManagement* which are allowed in the CMS.
- The required actions for the *PhoneManagement* which are allowed in the CMS.

PhoneManagement is conditionally required if the *Phone Data Model* is supported by the *TelephonyServer*.

5.1 Service Type

This specification reuse the *ConfigurationManagement* service exactly as specified in [7] by the UPnP Device Management WC:

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

PhoneManagement defined in this specification refers to the same service type.

5.2 *PhoneManagement* Architecture

The *PhoneManagement* is used by the *TelephonyServer* (TS) device to offer the management of the parameters defined in the *Phone Data Model* by profiling the CMS.

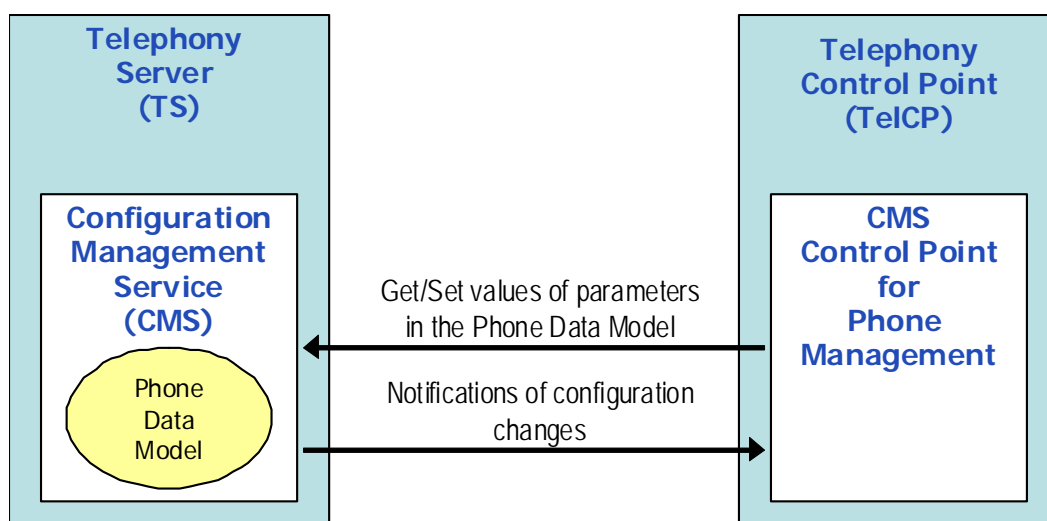The internal architecture of the *PhoneManagement* is in line with the Telephony architecture.

Figure 1 — *PhoneManagement* via CMS and *Phone Data Model*

5.2.1 Phone Management

The *PhoneManagement* is a set of procedures that implements the following general requirements of the UPnP Telephony:

- Managing the configuration of parameters like:

- Specific properties of the *TelephonyServer* (e.g. to manage capabilities related to the interfaces of a telephone like GSM, UMTS etc).

- Relevant concepts of the *TelephonyServer* to other Telephony services (e.g. the *AddressBook*).

- Retrieving the values of configuration parameters, status parameters for the diagnostics or troubleshooting purpose, and the counters for the monitoring (e.g. the current power supply source or the telephone battery status).

- Receiving the notifications of configuration parameters updates.

To fulfil the above requirements a TelCP can invoke the CMS actions for:

- Discovering the supported data model.

- Reading the current value of the configuration and status parameters.

- Managing the current configuration parameters for e.g. updating existing values in the *Phone Data Model*, creating new values in the *Phone Data Model*, or deleting the existing values from the *Phone Data Model*.

Any updates in the values of *Phone Data Model* parameters will be evented via a state variable.

5.2.2 Phone Data Model

The *Phone Data Model* defines a structure of list of configuration *Parameters* for the *PhoneManagement* in accordance to the rules specified by the CMS. The *Phone Data Model* includes the following parameters and information:

- Parameters for the *Address Book* in a form of a table, where each row corresponds to a contact with the associated relevant properties like name, mobile number, address, etc. The properties of a contact can be read or changed, the new contact can be added or existing contact can be deleted from the *Address Book*. The *Address Book* is the required feature in the *PhoneManagement* service.

- Parameters associated with an *answering machine* for setting the voice messages and to manage voice message items.

- Parameters for managing the *ringing alert* in the phone like *ringing mode (melody/vibrator or both)*, ring tones (melody).

- Parameters for reading the *GPS information* from a GPS receiver equipped in a telephone

- Parameter for configuring the *telephone profiles*, which are a set of meta information related to look and feel of a phone, service preferences (online, offline etc), and including personalized ring tones settings

- Parameters for reading the current *operational status* of the phone and for changing the administrative status (for e.g. enabling / disabling the telecommunication capabilities) of the phone.

- Parameters for tracking the level and status of the *battery* of a telephone.

- Personal profile information: User's own information for e.g. contact information, status image etc. which is also known as Personal Contact Card (PCC)

- Parameters for service settings and service settings profile for the WAN side services like Messaging, and Call.

- Parameters for the *Telephony* related settings and the profile for Telephony services like *Messaging* Service and *CallManagement* Service

- Connected home device information and device capability information like hardware capabilities, software capabilities of the device etc.

5.2.2.1   Attributes of *Phone Data Model Parameters*

The *Phone Data Model* is defined in compliance with the data models of CMS. The CMS have defined different attributes of the parameters in the data models. The *PhoneManagement* profiles these attributes based on the requirements by the *Phone Data Model*.

In compliance with the CMS, the following attributes shall be supported by all CMS implementations (see [7] for further details about their meaning, usage and constraints):

- *Type*: describes the type of a *Parameter*, making use of a limited subset of the SOAP data types.

- *Access*: describes the level of accessibility of a particular *Parameter*, e.g. specifies whether a TelCP can change the value of a *Parameter* or not, whether TelCP can create new *Instance*, or can delete a particular *Instance Node*.

The following attributes are allowed for the CMS, but these attribute shall be supported by *PhoneManagement*:

- *EventOnChange*: indicates whether to update the *ConfigurationUpdate* state variable when the *Parameter* or the *MultiInstance Node* in the data model is changed (and therefore the event shall be generated). The *Phone Data Model* in Annex A defines the list of *Parameters* which shall support *EventOnChange* attribute and their default value.

In order to provide some more sophisticated behavior, the *PhoneManagement* may support the following allowed attribute:

- *Version:* when a *Parameter* supporting this attribute changes its *Parameter* value (and the *CurrentConfigurationVersion* state variable is consequently updated), the *Version* attribute for such *Parameter* assumes the value of *CurrentConfigurationVersion*. (details can be found in [7], 2.4.22, Relationships Between State Variables). If the TS supports this attribute, is conditionally required to support also the *GetSelectedValues()* allowed action.

5.3   State Variables

The *PhoneManagement* reuses some of the state variables defined in the *ConfigurationManagement* service.

The Table 1 below lists all the eventable state variables used in the *PhoneManagement*. These state variables are defined in the CMS. The table also indicates the required/allowed constraint for the *PhoneManagement*.

All the required state variables from the CMS shall be implemented by *PhoneManagement*, as specified in [1]. It is up to the implementation to choose whether to support allowed state variables.

The *AttributeValuesUpdate* state variable should be supported if the CMS implementation support the *SetAttributeValues()* for changing attribute values (i.e. the values of *EventOnChange* and *Version*, when it is also supported).

The *PhoneManagement* supports the A_ARG_TYPE state variables defined in the CMS depending on the required/allowed constraint of the respective actions for the

PhoneManagement. The A_ARG_TYPE_ state variables are not listed in this document, refer to the [7] for the details.

Table 1 — State Variables for Eventing

Variable Name	CMS R/A a	TS R/A a
ConfigurationUpdate	*R*	*R*
CurrentConfigurationVersion	*R*	*R*
SupportedDataModelsUpdate	*R*	*R*
SupportedParametersUpdate	*R*	*R*
AttributeValuesUpdate	*A*	*A*
InconsistentStatus	*A*	*A*
a *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*).		

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.

5.4   Actions

5.4.1   Introduction

The *PhoneManagement* reuses the actions defined in the CMS. Some of the actions are allowed in CMS but those are required or conditionally required for *PhoneManagement*.

Table 2 below lists all the CMS actions, with the following additional information:

- The column "CMS R/A" indicates whether the action is required or allowed in [7]; all the required actions from the CMS shall be implemented by the device supporting *PhoneManagement*.

- The column "TS R/A" indicates whether the action is either required or allowed for *PhoneManagement*; some of the allowed actions are required for the *PhoneManagement*.

- The column "TelCP R/A" indicates whether the TelCP shall support the action to be fully compliant with the TS.

Refer to [7] for more detailed information on each action. The remaining subclauses of 5.4 give additional information on some of the actions as used by the *PhoneManagement* profile.

Table 2 — Actions

Name	CMS R/A a	TS R/A b	TelCP R/A c
GetSupportedDataModels()	*R*	*R*	*R*
GetSupportedParameters()	*R*	*R*	*R*
GetInstances()	*R*	*R*	*R*
GetValues()	*R*	*R*	*R*
GetSelectedValues()	*A*	*CR*	*CR*
SetValues()	*A*	*R*	*R*
CreateInstance()	*A*	*R*	*R*
DeleteInstance()	*A*	*R*	*R*

Name	CMS R/A [a]	TS R/A [b]	TelCP R/A [c]
GetAttributes()	*R*	*R*	*R*
SetAttributes()	*A*	*A*	*A*
GetInconsistentStatus()	*A*	*A*	*A*
GetConfigurationUpdate()	*R*	*R*	*R*
GetCurrentConfigurationVersion()	*R*	*R*	*R*
GetSupportedDataModelsUpdate()	*R*	*R*	*R*
GetSupportedParametersUpdate()	*R*	*R*	*R*
GetAttributeValuesUpdate()	*A*	*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 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*).

[c] For a control point 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*).

5.4.2   *GetSelectedValues()*

If *PhoneManagement* implementation supports the *Version* attribute in the *Phone Data Model*, then the *GetSelectedValues()* action shall be supported.

The *Version* attribute allows a TS to indicate a TelCP about the change in the respective *Parameter* value, or *MultiInstance Node* value. The *GetSelectedValues()* action is used to retrieve the actual change in the *Parameter* value or *MultiInstance Node* value. Refer to B.3.3 for detail mechanism for retrieving the changes.

5.4.3   *SetValues()*

The *SetValues()* action is an allowed action for the CMS, but the *PhoneManagement* perspective it is a required action and shall be implemented by all the implementation of the *PhoneManagement*.

This action is required because the *Phone Data Model* supports *Parameters* with write access permission. This action allows a TelCP to change the values of such *Parameters*. For the examples on the *Parameter* change refer to Annex B.

5.4.4   *CreateInstance()*

The *CreateInstance()* action is an allowed action for the CMS, but the *PhoneManagement* perspective it is a required action and shall be implemented by all the implementation of the *PhoneManagement*.

The *Phone Data Model* in the *PhoneManagement* supports *MultiInstance Nodes*, having write access permission e.g. *Address Book*. This action allows a TelCP to create a new *Instance* in the *Phone Data Model,* e.g. creating a new entry for a particular contact in the *Address Book*. For the example on the creating *Instances* in the *Phone Data Model* refer to Annex B.

5.4.5 *DeleteInstance()*

The *DeleteInstance()* action is an allowed action for the CMS, but the *PhoneManagement* perspective it is a required action and shall be implemented by all the implementation of the *PhoneManagement*.

The *Phone Data Model* includes some *MultiInstance Nodes* information which can be deleted by a TelCP, e.g. contact entries in the *Address Book.* This action therefore allows a TelCP to delete existing *Instance Nodes* from the *Phone Data Model*. For the example on the deleting an *Instance* in the *Phone Data Model* refer to Annex B.

5.4.6 *SetAttributes()*

The *SetAttributes()* action is an allowed action for the CMS, but the *PhoneManagement* perspective it is a conditionally required action and should be implemented by all the implementation of the *PhoneManagement*.

The *PhoneManagement* defines some of the allowed *Parameters* which requires *EventOnChange* attribute. This action allows a TelCP to enable or disable a *ConfigurationUpdate* event for those particular *Parameters*. If *PhoneManagement* implementation supports such *Parameters* then this action shall be supported in the implementation. Examples can be found in Annex B.

6 XML Service Description

The service description is specified in [7].

Annex A
(normative)

Phone Data Model

A.1   Introduction

Annex A defines the data model for the *TelephonyServer* device called as *Phone Data Model*.

The *Phone Data Model* defines the set of *Parameters* which are defined in compliance with the rules for data model definition specified in the CMS. The *Phone Data Model* does not require any support of the *Common Objects* defined in [7].

The following rules are defined in accordance with [7]:

- All *Parameter* names defined in Table A.1 shall be prefixed by /UPnP/PHONE/.

- Vendor specific *Parameters* to extend the capabilities of *Phone Data Model* shall begin with /…/X_ concatenated by the vendor domain name. For example: /UPnP/PHONE/X_MyCompany/ should be the prefix for all MyCompany specific parameters.

- The list of supported data models returned by *CMS::GetSupportedDataModels()* action (see *A_ARG_TYPE_SupportedDataModels* in [7]) shall include a row with the following information:

  - **URI**: `urn:UPnP:phone:2:pdm:2`.

  - **Location**: `/UPnP/PHONE/`.

  - **URL**: http://www.upnp.org/specs/phone/UPnP-phone-PhoneManagement-v2-Service.pdf.

  - **Description**: `Telephony: phone data model`.

  - **SourceLocation**: Not applicable.

Column descriptions in Table A.1 are identical to those given in [7], B.3, with the following addition:

In column *EOC*, a dash "-" means that the support for that attribute is allowed. The "0" means that the support for that attribute is required and its default value should be 0 (false). The "1" means that the support for that attribute is required and its default value should be 1 (true).

A.2   MIMEType and Value Management

The `MIMEType` and `Value` are a pair of *Parameters* which is used quite intensively in the following data model. They allow the management of Multi Media resources (e.g.: images, ring tones, voice messages and so on), whereas:

- `MIMEType`: describes `Value`'s content type using IANA registered formats. The `text/plain MIMEType` shall be used when `Value` contains a reference to the resource (URI). Other pecific IANA registered formats shall be used when `Value` contains the string encoded resource (e.g.: image/jpg for encoded jpg images).

- `Value`: contains either a reference to the resource (URI) or the resource value (properly encoded to be stored in a string type parameter). In case `Value` contains the reference, setting, deleting or changing its value does not imply the management of the resource

itself. The management of the referenced resource is implementation dependant and is out of scope of this protocol (e.g.: a new reference might be created for a non existing resource). In case `Value` contains the encoded value of the resource, setting, deleting or changing its value therefore implies setting, deleting or changing the resource itself.

Furthermore, since `MIMEType` and `Value` can be separately set by the TelCP, and there is no syntax check to their values, the set operation can be done in a not consistent way, for example the `MIMEType` might be set to text/plain and `Value` might contain some string encoded audio/video data. It's up to the PHONE device implementation to check and manage possible inconsistencies. For example, an extremely smart application might try to adapt the right `MIMEType` to the `Value` provided, whereas a simple application might simply return an error to the TelCP.

A.3   *Phone Data Model* Overview

Figure A.1 gives a structure overview of the *Phone Data Model* including *SingleInstance* (e.g. Identification), *MultiInstance* (e.g. AddressBook) and *InstanceAlias* (i.e. the '#' symbol) *Nodes*. *Parameters* are not included for sake of readability.
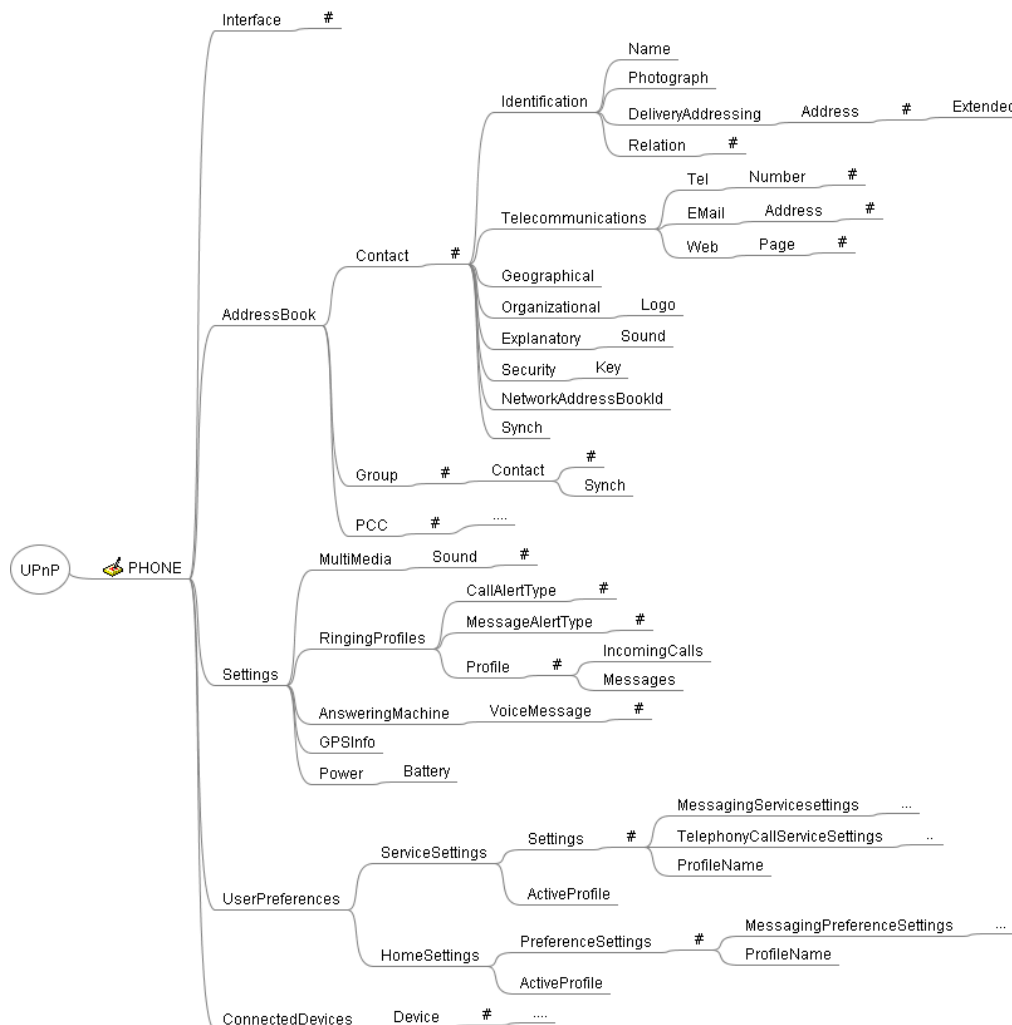
Figure A.1 — Overview of the *Phone Data Model*

Table A.1 — The *Phone Data Model*

Name	Type	Acc.	Req	Description	EOC	Ver
/UPnP/PHONE/	*SingleInstance*	-	R	The root node of the Phone Data Model.	-	-
InterfaceNumberOfEntries	unsignedInt	-	CR	Number of interfaces. It shall be provided when the /UPnP/PHONE/Interface/#/ is supported.	O	R
/UPnP/PHONE/Interface/#/	*MultiInstance*	-	A	Table that lists all the available "interfaces" on the Telephony Server, both at network transmission level and at telephony connectivity level, in order to allow status check and administration tasks (for enabling and disabling the single interface).	-	-
Name	string(64)	-	R	Name of this interface, as provided by the Telephony Server.	-	-
Enable	boolean	W	R	Command parameter in order to Enable (1) or Disable (0) this interface. The current value indicates if this interface it is Enabled (1) or Disabled (0). Disabling and then enabling an interface causes the restart of all that interface and all the supported connections.	O	R
Status	string	-	R	Current operational status of this interface. Some of the allowed values are: • "Initializing" • "Registering" • "Active" • "Unregistering" • "Error" • "Testing" • "Quiescent" • "Disabled"	-	-

Name	Type	Acc.	Req	Description	EOC	Ver
Type	string	-	R	The type of this interface. Some of the allowed values are: • "GSM/GPRS" • "UMTS/WCDMA" • "HSPA" • "WiFi" • "Bluetooth" • "USB" • "Ethernet" • "PSTN/ISDN" • "VoIP"	-	-
StartTime	dateTime	-	R	The absolute dateTime of the last successful restart of this interface.	-	-
/UPnP/PHONE/AddressBook/	SingleInstance	-	R	Container of Address Book management parameters	-	-
MaxContactsSupported	unsignedInt	-	R	This parameter declares the maximum number of contacts (entries, rows) that the Address Book can contain. If there is no upper limit, the value declared shall be 0.	0	-
MaxGroupsSupported	unsignedInt	-	R	This parameter declares the maximum number of groups (entries, rows) that the Address Book can contain. If there is no upper limit, the value declared shall be 0.	0	-
StorageLocation	string	-	R	It defines whether the Address Book is locally stored on the TS or other options. Possible values are the following: "local" "remote"	0	-
ContactNumberOfEntries	unsignedInt	-	R	Number of contacts in the AddressBook.	0	R
GroupNumberOfEntries	unsignedInt	-	CR	Number of groups. It shall be provided when the /UPnP/PHONE/AddressBook/Group/#/ is supported.	0	-

Name	Type	Acc.	Req	Description	EOC	Ver
/UPnP/PHONE/AddressBook/Contact/#/	*MultiInstance*	W	R	A table where each row is a contact. A contact can be created, updated, deleted, read. Each contact (row) shall be defined in terms of its properties (i.e. parameters): vCard specification [8] is a source for such properties. Each contact in this table of contacts is uniquely identified by the *Instance* identifier (see the primary key description in ConfigurationManagement:1 Service). . In case one or more of the following values is not defined, the empty string shall be used to represent "NO value defined". Constraints are specified in specific parameters' description, when needed.	-	-
NetworkAddressBookId	*String*	W	A	Element specifies the network address book with which this contact is associated. For e.g. GMAIL or FACEBOOK etc. If the contact belongs to multiple network address book, then this element includes all the identifiers of the network address books as a comma separated values. When setting this parameter the TS implementation should to update the corresponding network address books.	0	-
/UPnP/PHONE/AddressBook/Contact/#/Identification/	*SingleInstance*	-	R	This *SingleInstance* is used in the table of contacts to capture information associated with the identification and naming of this contact.	-	-
FormattedName	string(64)	R	R	Specifies the formatted text corresponding to the name of the contact. For example: Mr. John Q. Public\, Esq. Note that the COMMA character (ASCII decimal 44) is escaped as well as the "n" for the newline. [From [8]] In the vCard this field is assumed to be writeable. In the Phone Data Model, this parameter is populated by the TS according to the details provided in the /Identification/Name/ parameters. The TS rule for populate this parameter is out of scope.	0	-
NickName	string(64)	W	R	Specifies the text corresponding to the nickname of the contact. The nickname is the descriptive name given instead of or in addition to the one belonging to a person, place, or thing. It can also be used to specify a familiar form of a proper name specified by the FormattedName or Name. One or more text values separated by a COMMA character (ASCII decimal 44). For example: Jim,Jimmie. [From [8]]	0	-
Birthday	dateTime	W	A	Specifies the birth date of the contact.	0	-

Name	Type	Acc.	Req	Description	EOC	Ver
Anniversary	dateTime	W	A	Anniversary for the contact.	0	-
FileAs	string(64)	W	A	The name used to save the contact. This should be implemented by devices that need specific file names to identify the contact in the file system. Constraints on this parameter are completely implementation dependent. This parameter is an Microsoft Outlook extension to vCard specification [8].	0	-
Note	string(256)	W	A	Free text for notes.	0	-
RelationNumberOfEntries	unsignedInt	-	CR	Number instances in the Relation table. It shall be provided when the /UPnP/PHONE/AddressBook/Contact/#/Identification/Relation/#/ is supported.	0	-
/UPnP/PHONE/AddressBook/Contact/#/Identification/Name/	*SingleInstance*	-	R	This *SingleInstance* contains a collection of components of the name of the contact. Individual text components (e.g. GivenName) can include multiple text values separated by the comma character (ASCII decimal 44). For example: Philip,Paul.	-	-
FamilyName	string(64)	W	R	The family name of the contact. [From [8]]	0	-
GivenName	string(64)	W	R	The given name of the contact. [From [8]]	0	-
MiddleName	string(64)	W	A	The additional (middle) name of the contact.	0	-
AdditionalName	string(64)	W	A	The additional name of the contact. [From [8]]	0	-
HonorificPrefix	string(64)	W	A	The honorific prefix. For example: Mr, Dr, …[From [8]]	0	-
HonorificSuffix	string(64)	W	A	The honorific suffix. For example: Jr, Esq,….[From [8]]	0	-
/UPnP/PHONE/AddressBook/Contact/#/Identification/Relation/#/	*MultiInstance*	-	A	This *MultiInstance* contains a list of relations from the current contact to another contact in the *Address Book*, when there is any.	-	-

Name	Type	Acc.	Req	Description	EOC	Ver
Type	string	W	R	Values for this parameter are (the quotes shall not be used when specifying the values for this parameter): "Assistant", "Spouse", "Child", "Parent", "Familiar", "Friend", "Colleague", "Other". The Type is not required to be unique among the Instances. For example, in case there are two assistants, it is possible to have two different Instances having "Assistant" for Type value.	0	-
Note	string(256)	W	R	Free text for notes: other types of relations, names, addresses and so on.	0	-
ContactIdentifier	unsignedInt	W	R	The *Instance* identifier of the related contact: If there is a contact in this *Address Book* which represents the desired relation, this ContactIdentifier shall be the InstanceIdentifier of such contact. If there is no contact in this *Address Book* which represents the desired relation, this ContactIdentifier shall be 0 (zero).	0	-
/UPnP/PHONE/AddressBook/Contact/#/Identification/Photograph/	*SingleInstance*	-	A	An image or photograph information that annotates some aspect of the contact.	-	-
MIMEType	string	W	R	The type of the following Value parameter. Allowed values, as a subset from the IANA registered formats http://www.iana.org/assignments/media-types/, are: • text/plain (http://www.iana.org/assignments/media-types/text/). • all image/… media types (http://www.iana.org/assignments/media-types/image/). [From [8]]	0	-

Name	Type	Acc.	Req	Description	EOC	Ver
Value	string(32k)	W	R	The "value" of the photograph, depending on the MIMEType: • If the MIMEType is text/plain this parameter shall contain the URI where the photograph is defined. For example: http://www.abc.com/pub/photos/jqpublic.gif • If the MIMEType is image/... this parameter shall contain the photograph encoded with respect to the image MIME type specified. For example, if the MIMEType is image/jpg the Value shall contain the encoded jpg image. [From [8]]	0	-
/UPnP/PHONE/AddressBook/Contact/#/DeliveryAddressing/	*SingleInstance*	-	A	Specifies the components of the delivery address for the contact. This *SingleInstance* contains a table of address components (e.g. postal code, city,...).	-	-
Preferred	unsignedInt	W	R	The *Instance* identifier of the preferred address in the following list of Addresses. This shall be an existing value. • When there are NO Instances of Addresses, the value is 0 and can not be changed. An attempt to change that value shall return the proper fault code (CMS::705 "Invalid Value"). • When there is only one *Instance* of Addresses, the value is the *Instance* identifier of such *Instance*. An attempt to set invalid values will result in the proper fault code returned by the device (CMS::705 "Invalid Value"). • When there are more than a single *Instance* of Addresses, it's up to the implementation to define the default value, that can be also updated by a TelCP. An attempt to set invalid values will result in the proper fault code returned by the device (CMS::705 "Invalid Value"). For example, if there are the *Instance* identifier 3, 5 and 67, the control point can set this parameter to 3, 5 or 67. Any other values will produce an error. [From [8]]	0	-
AddressNumberOfEntries	unsignedInt	-	R	Number of delivery addresses for this contact.	0	-
/UPnP/PHONE/AddressBook/Contact/#/DeliveryAddressing/Address/#/	*MultiInstance*	W	R	This *MultiInstance* contains the addresses components of the contact. Where it makes semantic sense, individual text components of the following Address *MultiInstance* children can include multiple text values (e.g., The "Street" parameter with multiple lines) separated by the COMMA character (ASCII decimal 44).	-	-

Name	Type	Acc.	Req	Description	EOC	Ver
Type	string	W	R	Possible values for this parameter are (the quotes shall not be used when specifying the values for this parameter): • "dom" to indicate a domestic delivery address; • "intl" to indicate an international delivery address; • "postal" to indicate a postal delivery address; • "parcel" to indicate a parcel delivery address; • "home" to indicate a delivery address for a residence; • "work" to indicate delivery address for a place of work; • "pref" to indicate the preferred delivery address when more than one address is specified. The value of this parameter shall be at least (see below) one of the above types, and shall NOT be the empty string. An attempt to set invalid values will result in the proper fault code returned by the device (CMS::705 "Invalid Value"). The Type is not required to be unique among the Instances. For example, in case there are two different working places, it is possible to have two different Instances having "work" for Type value. The value for this parameter can be specified as well as a list of values (i.e., " dom, postal") separated by the COMMA character (ASCII decimal 44). For example, a domestic delivery address for postal delivery to a residence that is also used for work will be "dom,postal,work,home". [From [8]]	0	-
PostOfficeBox	string(64)	W	A	The post office box. [From [8]]	0	-
ExtendedAddress	string(64)	W	A	The extended address. [From [8]]	0	-
StreetAddress	string(64)	W	R	The street address. [From [8]]	0	-
Locality	string(64)	W	R	The name of the locality (e.g. village or city). [From [8]]	0	-
Region	string(64)	W	R	The name of the region (e.g. state or province) . [From [8]]	0	-
PostalCode	string(16)	W	R	The ZIP or postal code. [From [8]]	0	-
Country	string(64)	W	R	The country name (e.g. U.S.A.). [From [8]]	0	-

Name	Type	Acc.	Req	Description	EOC	Ver
Label	string(512)	W	R	Specifies the formatted text corresponding to delivery address of the contact. Contains the formatted text that can be used to present a delivery address label for the related Addresses *Instance*. For example, if the FormattedName is "", the StreetAddress is "" and the Country is "", the Label value might be a multi-line address string as follows: "Mr.John Q. Public\n123 Main Street\nU.S.A" Note that the COMMA character (ASCII decimal 44) is escaped as well as the "n" for the newline. [From [8]]	0	-
/UPnP/PHONE/AddressBook/Contact/#/DeliveryAddressing/Address/#/Extended/	*SingleInstance*	-	A	Extended information to the address. [From [8]]	-	-
Building	string(64)	W	R	Name of the building, if any.	0	-
Office	string(64)	W	R	Office identification, if any.	0	-
Room	string(64)	W	R	Room number or name, if any.	0	-
Floor	string(64)	W	R	Floor number or name, if any.	0	-
/UPnP/PHONE/AddressBook/Contact/#/Telecommunications/	*SingleInstance*	-	R	This *SingleInstance* concerns with information associated with the telecommunications addressing of the contact.	-	-
/UPnP/PHONE/AddressBook/Contact/#/Telecommunications/Tel/	*SingleInstance*	-	R	This *SingleInstance* contains information associated with thecontact telephone numbers for telephony communication.	-	-
Preferred	unsignedInt	W	R	The *Instance* identifier of the preferred phone number in the following list of Numbers. This shall be an existing value. • When there are NO Instances of Number, the value is 0 and can not be changed. An attempt to change that value shall return the proper fault code (CMS::705 "Invalid Value"). • When there is only one *Instance* of Number, the value is the *Instance* identifier of such *Instance*. An attempt to set invalid values will result in the proper fault code returned by the device (CMS::705 "Invalid Value"). • When there are more than a single *Instance* of Number, it's up to the implementation to define the default value. An attempt to set invalid values will result in the proper fault code returned by the device (CMS::705 "Invalid Value"). For example, if there are the *Instance* identifier 3, 5 and 67, the control point can set this parameter to 3, 5 or 67. Any other values will produce an error. [From [8]]	0	R

Name	Type	Acc.	Req	Description	EOC	Ver
NumberNumberOfEntries	unsignedInt	-	R	Number of telephone Numbers in the following table.	-	-
/UPnP/PHONE/AddressBook/Contact/#/Telecommunications/Tel/Number/#/	*MultiInstance*	W	R	This *MultiInstance* contains the Tel components of the contact.	O	R
Type	string	W	R	Possible values for this parameter are (the quotes shall not be used when specifying the values for this parameter): "Mobile", "Mobile(home)", "Mobile(business)", "Telephone", "Telephone(home)", "Telephone(business)", "Video", "Video(home)", "Video(business)", "Internet", "Internet(home)", "Internet(business)", "FAX", "FAX(home)", "FAX(business)", "Car-phone", "Pager", "BBS" to indicate a bulletin board system telephone number, "Modem" to indicate a MODEM connected telephone number, "ISDN" to indicate an ISDN service telephone number, "PCS" to indicate a personal communication services telephone number. The value of this parameter shall be at least (see below) one of the above types, and shall NOT be the empty string. An attempt to set invalid values will result in the proper fault code returned by the device (CMS::705 "Invalid Value"). The Type is not required to be unique among the Instances. For example, in case there are two different working places, both having the ISDN service, it is possible to have two different Instances having "ISDN" for Type value. [From [8]]	O	R
Country	string(4)	W	A	The country international prefix (e.g. "+1" for the U.S.A.). This parameter is an Outlook extension to vCard specification [8].	O	R
Prefix	string(4)	W	A	The City/Area code (e.g. 206). This parameter is an Outlook extension to vCard specification [8].	O	R
Number	string(64)	W	R	The local telephone number. [From [8]]	O	R
Extension	string(64)	W	A	The extension of the telephone number. This parameter is an Outlook extension to vCard specification [8].	O	-
/UPnP/PHONE/AddressBook/Contact/#/Telecommunications/EMail/	*SingleInstance*	-	R	Specifies the electronic mail addresses for communication with the contact.	-	-

Name	Type	Acc.	Req	Description	EOC	Ver
Mailer	string(64)	W	A	The type of electronic mail software that is used by the individual associated with the contact. For example: "PigeonMail 2.1". [From [8]]	0	-
Preferred	unsignedInt	W	R	The *Instance* identifier of the preferred mail address in the following list of addresses. This shall be an existing value. • When there are NO Instances of Address, the value is 0 and can not be changed. An attempt to change that value shall return the proper fault code (CMS::705 "Invalid Value"). • When there is only one *Instance* of Address, the value is the *Instance* identifier of such *Instance*. An attempt to set invalid values will result in the proper fault code returned by the device (CMS::705 "Invalid Value"). • When there are more than a single *Instance* of Address, it's up to the implementation to define the default value. An attempt to set invalid values will result in the proper fault code returned by the device (CMS::705 "Invalid Value"). For example, if there are the *Instance* identifier 3, 5 and 67, the control point can set this parameter to 3, 5 or 67. Any other values will produce an error. [From [8]]	0	R
WebAddress	string(256)	W	A	The web address. This parameter is an Outlook extension to vCard specification [8].	0	-
IMAddress	string(256)	W	A	Instant Messaging address. This parameter is an Outlook extension to vCard specification [8].	0	R
AddressNumberOfEntries	unsignedInt	-	R	Number of email addresses.	0	-
/UPnP/PHONE/AddressBook/Contact/#/Telecommunications/EMail/Address/#/	*MultiInstance*	W	R	This *MultiInstance* contains the e-mail addresses component of the contact.	0	R

Name	Type	Acc.	Req	Description	EOC	Ver
Type	string	W	R	Values for this parameter are (the quotes shall not be used when specifying the values for this parameter): "Home", the address is for the whole home/family use "Business", the address is used for business "Personal", the address is for private uses "Other", any other purposes fro this address The value of this parameter shall be at least (see below) one of the above types, and shall NOT be the empty string. An attempt to set invalid values will result in the proper fault code returned by the device (CMS::705 "Invalid Value"). The Type is not required to be unique among the Instances. For example, in case there are two different business e-mail addresses, it is possible to have two different Instances having "Business" for Type value.	0	R
Address	string(256)	W	R	The internet e-mail address. [From [8]]	0	R
DisplayAs	string(64)	W	A	The formatted name used to display the address. For example, if the e-mail address is related to work, it might be desirable to display it as "work address". This parameter is an Outlook extension to vCard specification [8].	0	-
/UPnP/PHONE/AddressBook/Contact/#/Telecommunications/Web/	*SingleInstance*	-	A	Specifies the Web pages associated with the contact.	-	-
Preferred	unsignedInt	W	R	The *Instance* identifier of the preferred URL in the following list of Web Pages. This shall be an existing value. • When there are NO Instances of Pages, the value is 0 and can not be changed. An attempt to change that value shall return the proper fault code (CMS::705 "Invalid Value"). • When there is only one *Instance* of Pages, the value is the *Instance* identifier of such *Instance*. An attempt to set invalid values will result in the proper fault code returned by the device (CMS::705 "Invalid Value"). • When there are more than a single *Instance* of Pages, it's up to the implementation to define the default value. An attempt to set invalid values will result in the proper fault code returned by the device (CMS::705 "Invalid Value"). For example, if there are the *Instance* identifier 3, 5 and 67, the control point can set this parameter to 3, 5 or 67. Any other values will produce an error.	0	-
PageNumberOfEntries	unsignedInt	-	R	Number of web pages.	0	-

Name	Type	Acc.	Req	Description	EOC	Ver
/UPnP/PHONE/AddressBook/Contact/#/Telecommunications/Web/Page/#/	MultiInstance	W	R	This *MultiInstance* contains the Web Pages component of the contact.	-	-
Type	string	W	R	Values for this parameter are (the quotes shall not be used when specifying the values for this parameter): "Home", the address is for the whole home/family use "Business", the address is used for business "Personal", the address is for private uses "Other", any other purposes fro this address The value of this parameter shall be at least (see below) one of the above types, and shall NOT be the empty string. An attempt to set invalid values will result in the proper fault code returned by the device (CMS::705 "Invalid Value"). The Type is not required to be unique among the Instances. For example, in case there are two different business web pages, it is possible to have two different Instances having "Business" for Type value.	0	-
URL	string(256)	W	A	The Web address. It shall be a valid URL.	0	-
/UPnP/PHONE/AddressBook/Contact/#/Geographical/	*SingleInstance*	-	A	This *SingleInstance* is concerned with information associated with geographical positions or regions associated with the contact	-	-
TZ	string(64)	W	A	Specifies information related to the time zone of the contact. The default is a single UTC-offset value. It can also be reset to a single text value. For example: "05:00; EST; Raleigh/North America" [From [8]]	0	-

Name	Type	Acc.	Req	Description	EOC	Ver
Geo	string(64)	W	A	To specify information related to the global positioning of the contact. This type specifies information related to the global position of the contact. The value specifies latitude and longitude, in that order (i.e., "LAT LON" ordering). The longitude represents the location east and west of the prime meridian as a positive or negative real number, respectively. The latitude represents the location north and south of the equator as a positive or negative real number, respectively. The longitude and latitude values shall be specified as decimal degrees and should be specified to six decimal places. This will allow for granularity within a meter of the geographical position. The string components are separated by the SEMI-COLON character (ASCII decimal 59). The simple formula for converting degrees-minutes-seconds into decimal degrees is: decimal = degrees + minutes/60 + seconds/3600. This is an example of Geo value: "37.386013;-122.082932". [From [8]]	0	-
/UPnP/PHONE/AddressBook/Contact/#/Organizational/	*SingleInstance*	-	A	This *SingleInstance* is concerned with information associated with characteristics of the organization or organizational units of the contact.	-	-
Title	string(64)	W	R	The job title, functional position or function of the contact. For example: "Director\, Research and Development" Note that the COMMA character (ASCII decimal 44) is escaped with the back-slash \. [From [8]]	0	-
Role	string(64)	W	R	Specifies information concerning the role, occupation, or business category of the contact. This parameter is included as an organizational type to avoid confusion with the semantics of the Title parameter and incorrect usage of that type when the semantics of this parameter is intended. For example: "Programmer" [From [8]]	0	-

Name	Type	Acc.	Req	Description	EOC	Ver
Agent	string(64)	W	R	Specifies information about another person who will act on behalf of the individual or resource associated with the contact. A key characteristic of the Agent type is that it represents somebody or something that is separately addressable. Possible, and mutually exclusive value types for this parameter are: • A single vCard value (RFC 2426). For example "BEGIN:VCARD\nFN: ...END:VCARD\n". • A reference to a contact in this AddressBook. For Example: "/UPnP/PHONE/AddressBook/Contact/45/". This extends the to vCard specification [8]. • An URI to specify information outside of this scope. For example: "URI:CID:JQPUBLIC.part3.960129T083020.xyzMail@host3.com". [From [8]]	0	-
Org	string(64)	W	R	Specifies the organizational name and units. This is a string consisting of components separated the SEMI-COLON character (ASCII decimal 59). The first part is the organization name, followed by one or more levels of organizational unit names. For example:"ABC), Inc.;North American Division;Marketing". [From [8]]	0	-
/UPnP/PHONE/AddressBook/Contact/#/Organizational/Logo/	*SingleInstance*	-	A	Graphic image of a logo associated with the contact.	-	-
MIMEType	string	W	R	The type of the following Value parameter. Allowed values, as a subset from the IANA registered formats http://www.iana.org/assignments/media-types/, are: • text/plain (http://www.iana.org/assignments/media-types/text/). • all image/… media types (http://www.iana.org/assignments/media-types/image/). [From [8]]	0	-

Name	Type	Acc.	Req	Description	EOC	Ver
Value	string(32k)	W	R	The "value" of the logo, depending on the MIMEType: • If the MIMEType is text/plain this parameter shall contain the URI where the contact's logo is defined. For example: http://www.abc.com/pub/logos/jqpublic.gif • If the MIMEType is image/… this parameter shall contain the logo encoded with respect to the image MIME type specified. For example, if the MIMEType is image/jpg the Value shall contain the encoded jpg logo image. [From [8]]	0	-
/UPnP/PHONE/AddressBook/Contact/#/Explanatory/	*SingleInstance*	-	A	This *SingleInstance* contains additional explanations, such as that related to informational notes or revisions specific to the contact.	-	-
Categories	string	W	R	Specifies application category information about the contact description (e.g. the vCard). One or more text values separated by a COMMA character (ASCII decimal 44). For example: "TRAVEL AGENT" or "INTERNET,IETF,INDUSTRY,INFORMATION TECHNOLOGY" Are valid values for this parameter. [From [8]]	0	-
Note	string(256)	W	R	Specifies supplemental information or a comment that is associated with the contact description (e.g. the vCard). For example: "This fax number is operational 0800 to 1715 EST, Mon-Fri". [From [8]]	0	-
Prod	string(64)	W	A	Specifies the identifier for the product that created the contact information (e.g. the vCard). Implementations should use a method such as that specified for Formal Public Identifiers in ISO 9070 to assure that the text value is unique. For example: "-//ONLINE DIRECTORY//NONSGML Version 1//EN". [From [8]]	0	-
Rev	string(64)	W	A	Specifies revision information about the contact description (e.g. the vCard). The value distinguishes the current revision of the information in this AddressBook's contact for other renditions of the information. For example: "1995-10-31T22:27:10Z". [From [8]]	0	-

Name	*Type*	*Acc.*	Req	Description	*EOC*	*Ver*
SortString	string(64)	W	R	Specifies the family name or given name text to be used for national-language-specific sorting of the FormattedName and Name within /UPnP/PHONE/AddressBook/Contact/#/Identification/.	0	-
				The sort string is used to provide family name or given name text that is to be used in locale- or national-language-specific sorting of the formatted name and structured name types. Without this information, sorting algorithms could incorrectly sort this contact within a sequence of sorted contacts in the AddressBook.		
				When this parameter value is assigned to a contact, then this family name or given name value is used for sorting among contacts.		
				For example, in the case of family name sorting, the following examples define common sort string usage with the FormattedName and Name parameters.		
				/../FormattedName: Rene van der Harten		
				/../Name/FamilyName: van der Harten;		
				/../Name/GivenName:Rene		
				...		
				/../SortString: Harten		
				/../FormattedName:Robert Pau Shou Chang		
				/../Name/FamilyName:Pau		
				/../Name/GivenName: Shou Chang		
				...		
				/../SortString: Pau		
				/../FormattedName:Osamu Koura		
				/../Name/FamilyName:Koura		
				/../Name/GivenName:Osamu		
				...		
				/../SortString: Koura		
				[From [8]]		

Name	Type	Acc.	Req	Description	EOC	Ver
UID	string(256)	W	R	Specifies a value that represents a globally unique identifier corresponding to the individual or resource associated with the contact. This parameter value should be an IANA registered identifier format, but a non-standard format is anyway permitted. Example: "19950401-080045-40000F192713-0052". [From [8]]	0	-
URL	string(256)	W	R	Specifies a uniform resource locator associated with this contact. [From [8]]	0	-
/UPnP/PHONE/AddressBook/Contact/#/Explanatory/Sound/	SingleInstance	-	A	This SingleInstance is used to specify a digital sound content information that annotates some aspect of the contact. By default this type is used to specify the proper pronunciation of the name type value of contact.	-	-
MIMEType	string	W	R	The type of the following Value parameter. Allowed values, as a subset from the IANA registered formats http://www.iana.org/assignments/media-types/, are: • text/plain (http://www.iana.org/assignments/media-types/text/). • all audio/… media types (http://www.iana.org/assignments/media-types/audio/). [From [8]]	0	-
Value	string(32k)	W	R	The "value" of the sound, depending on the MIMEType: • If the MIMEType is text/plain this parameter shall contain the URI where the contact's sound is defined. For example: http://www.abc.com/pub/thiscontact/pronunciation.mpeg • If the MIMEType is audio/… this parameter shall contain the sound encoded with respect to the aoudio MIME type specified. For example, if the MIMEType is audio/mp4 the Value shall contain the encoded mp4 audio stream. [From [8]]	0	-
/UPnP/PHONE/AddressBook/Group/#/	MultiInstance	W	A	A table where each row is a group of contacts. A group can be created, updated, deleted, read.	0	R
Name	string(64)	W	R	Name of the group.	0	-
Address	string(256)	W	R	A valid URI used to identify the group address in the context of the application or service, out of scope of this specification. For example: "https://www112.livemeeting.com/cc/microsoft/join?id=NRCC5H&role=present&pw=any".	0	R

Name	Type	Acc.	Req	Description	EOC	Ver
Type	string	W	R	Values for this parameter are (the quotes shall not be used when specifying the values for this parameter): "Friends", "Familiars", "Parents", "Children", "Colleagues", "Organization", "Group", "Sport", "Hobby", "Other". The Type is not required to be unique among the Instances. For example, in case there are two different groups, both related to friends, it is possible to have two different Instances having "Friends" for Type value.	0	-
Note	string(256)	W	R	Free text for notes: other types of groups, names, addresses and so on.	0	-
ContactNumberOfEntries	unsignedInt	-	R	Number of contacts belonging to this group.	0	-
/UPnP/PHONE/AddressBook/Group/#/Contact/#/	MultiInstance		R	A table where each row is a contact belonging to the group. A contact for the group can be created, updated, deleted, read.	0	R
ContactIdentifier	unsignedInt	W	R	The Instance identifier of the contact belonging to the group. This shall be an existing valid contact. If the related contact is deleted, this Instance shall be deleted.	0	R
/UPnP/PHONE/AddressBook/PCC/	SingleInstance	-	R	This stores the PCC information for the user	O	-
NetworkAddressBookId	String	W	A	Element specifies the network address book with which this PCC information is associated. For e.g. GMAIL or FACEBOOK etc. If the PCC information belongs to multiple network address book, then this element includes all the identifiers of the network address books as a comma separated values. When setting this parameter the TS implementation should to update the corresponding network address books.	O	-
/UPnP/PHONE/AddressBook/PCC/Identification/	SingleInstance	-	R	This SingleInstance is used in the table of contacts to capture information associated with the identification and naming of this contact.	-	-

Name	Type	Acc.	Req	Description	EOC	Ver
FormattedName	string(64)	W	R	Specifies the formatted text corresponding to the name of the contact. For example: Mr. John Q. Public\, Esq. Note that the COMMA character (ASCII decimal 44) is escaped as well as the "n" for the newline. [From [8]] In the vCard this field is assumed to be writeable. In the Phone Data Model, this parameter is populated by the TS according to the details provided in the /Identification/Name/ parameters. The TS rule for populate this parameter is out of scope.	0	-
NickName	string(64)	W	R	Specifies the text corresponding to the nickname of the contact. The nickname is the descriptive name given instead of or in addition to the one belonging to a person, place, or thing. It can also be used to specify a familiar form of a proper name specified by the FormattedName or Name. One or more text values separated by a COMMA character (ASCII decimal 44). For example: Jim,Jimmie. [From [8]]	0	-
Birthday	dateTime	W	A	Specifies the birth date of the contact.	0	-
Anniversary	dateTime	W	A	Anniversary for the contact.	0	-
FileAs	string(64)	W	A	The name used to save the contact. This should be implemented by devices that need specific file names to identify the contact in the file system. Constraints on this parameter are completely implementation dependent. This parameter is an Outlook extension to vCard specification [8].	0	-
Note	string(256)	W	A	Free text for notes.	0	-
RelationNumberOfEntries	unsignedInt	-	CR	Number instances in the Relation table. It shall be provided when the /UPnP/PHONE/AddressBook/Contact/#/Identification/Relation/#/ is supported.	0	-
/UPnP/PHONE/AddressBook/PCC/#/Identification/Name/	SingleInstance	-	R	This SingleInstance contains a collection of components of the name of the contact. Individual text components (e.g. GivenName) can include multiple text values separated by the comma character (ASCII decimal 44). For example: Philip,Paul.	-	-
FamilyName	string(64)	W	R	The family name of the contact. [From [8]]	0	-
GivenName	string(64)	W	R	The given name of the contact. [From [8]]	0	-
MiddleName	string(64)	W	A	The additional (middle) name of the contact.	0	-

Name	Type	Acc.	Req	Description	EOC	Ver
AdditionalName	string(64)	W	A	The additional name of the contact. [From [8]]	0	-
HonorificPrefix	string(64)	W	A	The honorific prefix. For example: Mr, Dr, ...[From [8]]	0	-
HonorificSuffix	string(64)	W	A	The honorific suffix. For example: Jr, Esq.,... [From [8]]	0	-
/UPnP/PHONE/AddressBook/PCC/Identification/Photograph/	*SingleInstance*	-	A	An image or photograph information that annotates some aspect of the contact.	-	-
MIMEType	string	W	R	The type of the following Value parameter. Allowed values, as a subset from the IANA registered formats http://www.iana.org/assignments/media-types/, are: • text/plain (http://www.iana.org/assignments/media-types/text/). • all image/... media types (http://www.iana.org/assignments/media-types/image/). [From [8]]	0	-
Value	string(32k)	W	R	The "value" of the photograph, depending on the MIMEType: • If the MIMEType is text/plain this parameter shall contain the URI where the photograph is defined. For example: http://www.abc.com/pub/photos/jqpublic.gif • If the MIMEType is image/... this parameter shall contain the photograph encoded with respect to the image MIME type specified. For example, if the MIMEType is image/jpg the Value shall contain the encoded jpg image. [From [8]]	0	-
/UPnP/PHONE/AddressBook/PCC/DeliveryAddressing/	*SingleInstance*	-	A	Specifies the components of the delivery address for the contact. This *SingleInstance* contains a table of address components (e.g. postal code, city,...).	-	-

31

Name	Type	Acc.	Req	Description	EOC	Ver
Preferred	unsignedInt	W	R	The *Instance* identifier of the preferred address in the following list of Addresses. This shall be an existing value. • When there are NO Instances of Addresses, the value is 0 and can not be changed. An attempt to change that value shall return the proper fault code (CMS::705 "Invalid Value"). • When there is only one *Instance* of Addresses, the value is the *Instance* identifier of such *Instance*. An attempt to set invalid values will result in the proper fault code returned by the device (CMS::705 "Invalid Value"). • When there are more than a single *Instance* of Addresses, it's up to the implementation to define the default value, that can be also updated by a TelCP. An attempt to set invalid values will result in the proper fault code returned by the device (CMS::705 "Invalid Value"). For example, if there are the *Instance* identifier 3, 5 and 67, the control point can set this parameter to 3, 5 or 67. Any other values will produce an error. [From [8]]	0	-
AddressNumberOfEntries	unsignedInt	-	R	Number of delivery addresses for this contact.	0	-
/UPnP/PHONE/AddressBook/PCC/DeliveryAddressing/Address/#/	*MultiInstance*	R	R	This *MultiInstance* contains the addresses components of the contact. Where it makes semantic sense, individual text components of the following Address *MultiInstance* children can include multiple text values (e.g., The "Street" parameter with multiple lines) separated by the COMMA character (ASCII decimal 44).	-	-

Name	Type	Acc.	Req	Description	EOC	Ver
Type	string	W	R	Possible values for this parameter are (the quotes shall not be used when specifying the values for this parameter): • "dom" to indicate a domestic delivery address; • "intl" to indicate an international delivery address; • "postal" to indicate a postal delivery address; • "parcel" to indicate a parcel delivery address; • "home" to indicate a delivery address for a residence; • "work" to indicate delivery address for a place of work; • "pref" to indicate the preferred delivery address when more than one address is specified. The value of this parameter shall be at least (see below) one of the above types, and shall NOT be the empty string. An attempt to set invalid values will result in the proper fault code returned by the device (CMS::705 "Invalid Value"). The Type is not required to be unique among the Instances. For example, in case there are two different working places, it is possible to have two different Instances having "work" for Type value. The value for this parameter can be specified as well as a list of values (i.e., " dom, postal") separated by the COMMA character (ASCII decimal 44). For example, a domestic delivery address for postal delivery to a residence that is also used for work will be "dom,postal,work,home". [From [8]]	0	-
PostOfficeBox	string(64)	W	A	The post office box. [From [8]]	0	-
ExtendedAddress	string(64)	W	A	The extended address. [From [8]]	0	-
StreetAddress	string(64)	W	R	The street address. [From [8]]	0	-
Locality	string(64)	W	R	The name of the locality (e.g. village or city). [From [8]]	0	-
Region	string(64)	W	R	The name of the region (e.g. state or province) . [From [8]]	0	-
PostalCode	string(16)	W	R	The ZIP or postal code. [From [8]]	0	-
Country	string(64)	W	R	The country name (e.g. U.S.A.). [From [8]]	0	-

Name	Type	Acc.	Req	Description	EOC	Ver
Label	string(512)	W	R	Specifies the formatted text corresponding to delivery address of the contact. Contains the formatted text that can be used to present a delivery address label for the related Addresses *Instance*. For example, if the FormattedName is "", the StreetAddress is "" and the Country is "", the Label value might be a multi-line address string as follows: "Mr.John Q. Public\n123 Main Street\nU.S.A" Note that the COMMA character (ASCII decimal 44) is escaped as well as the "n" for the newline. [From [8]]	0	-
/UPnP/PHONE/AddressBook/PCC/DeliveryAddressing/Address/#/Extended/	*SingleInstance*	-	A	Extended information to the address.	-	-
Building	string(64)	W	R	Name of the building, if any.	0	-
Office	string(64)	W	R	Office identification, if any.	0	-
Room	string(64)	W	R	Room number or name, if any.	0	-
Floor	string(64)	W	R	Floor number or name, if any.	0	-
/UPnP/PHONE/AddressBook/PCC/Telecommunications/	*SingleInstance*	-	R	This *SingleInstance* concerns with information associated with the telecommunications addressing of the contact.	-	-
/UPnP/PHONE/AddressBook/PCC/Telecommunications/Tel/	*SingleInstance*	-	R	This *SingleInstance* contains information associated with the contact telephone numbers for telephony communication.	-	-
Preferred	unsignedInt	W	R	The *Instance* identifier of the preferred phone number in the following list of Numbers. This shall be an existing value. • When there are NO Instances of Number, the value is 0 and can not be changed. An attempt to change that value shall return the proper fault code (CMS::705 "Invalid Value"). • When there is only one *Instance* of Number, the value is the *Instance* identifier of such *Instance*. An attempt to set invalid values will result in the proper fault code returned by the device (CMS::705 "Invalid Value"). • When there are more than a single *Instance* of Number, it's up to the implementation to define the default value. An attempt to set invalid values will result in the proper fault code returned by the device (CMS::705 "Invalid Value"). For example, if there are the *Instance* identifier 3, 5 and 67, the control point can set this parameter to 3, 5 or 67. Any other values will produce an error. [From [8]]	0	R
NumberNumberOfEntries	unsignedInt	-	R	Number of telephone Numbers in the following table.	-	-

Name	Type	Acc.	Req	Description	EOC	Ver
/UPnP/PHONE/AddressBook/ PCC/Telecommunications/Tel/Number/#/	*MultiInstance*	W	R	This *MultiInstance* contains the Tel components of the contact.	0	R
Type	string	W	R	Possible values for this parameter are (the quotes shall not be used when specifying the values for this parameter): "Mobile", "Mobile(home)", "Mobile(business)", "Telephone", "Telephone(home)", "Telephone(business)", "Video", "Video(home)", "Video(business)", "Internet", "Internet(home)", "Internet(business)", "FAX", "FAX(home)", "FAX(business)", "Car-phone", "Pager", "BBS" to indicate a bulletin board system telephone number, "Modem" to indicate a MODEM connected telephone number, "ISDN" to indicate an ISDN service telephone number, "PCS" to indicate a personal communication services telephone number. The value of this parameter shall be at least (see below) one of the above types, and shall NOT be the empty string. An attempt to set invalid values will result in the proper fault code returned by the device (CMS::705 "Invalid Value"). The Type is not required to be unique among the Instances. For example, in case there are two different working places, both having the ISDN service, it is possible to have two different Instances having "ISDN" for Type value. [From [8]]	0	R
Country	string(4)	W	A	The country international prefix (e.g. "+1" for the U.S.A.). This parameter is an Outlook extension to vCard specification [8].	0	R
Prefix	string(4)	W	A	The City/Area code (e.g. 206). This parameter is an Outlook extension to vCard specification [8].	0	R
Number	string(64)	W	R	The local telephone number. [From [8]]	0	R
Extension	string(64)	W	A	The extension of the telephone number. This parameter is an Outlook extension to vCard specification [8].	0	-
/UPnP/PHONE/AddressBook/PCC/Telecommunications/EMail/	*SingleInstance*	-	R	Specifies the electronic mail addresses for communication with the contact.	-	-

Name	Type	Acc.	Req	Description	EOC	Ver
Mailer	string(64)	W	A	The type of electronic mail software that is used by the individual associated with the contact. For example: "PigeonMail 2.1". [From [8]]	0	-
Preferred	unsignedInt	W	R	The Instance identifier of the preferred mail address in the following list of addresses. This shall be an existing value. • When there are NO Instances of Address, the value is 0 and can not be changed. An attempt to change that value shall return the proper fault code (CMS::705 "Invalid Value"). • When there is only one Instance of Address, the value is the Instance identifier of such Instance. An attempt to set invalid values will result in the proper fault code returned by the device (CMS::705 "Invalid Value"). • When there are more than a single Instance of Address, it's up to the implementation to define the default value. An attempt to set invalid values will result in the proper fault code returned by the device (CMS::705 "Invalid Value"). For example, if there are the Instance identifier 3, 5 and 67, the control point can set this parameter to 3, 5 or 67. Any other values will produce an error. [From [8]]	0	R
WebAddress	string(256)	W	A	The web address. This parameter is an Outlook extension to vCard specification [8].	0	-
IMAddress	string(256)	W	A	Instant Messaging address. This parameter is an Outlook extension to vCard specification [8].	0	R
AddressNumberOfEntries	unsignedInt	-	R	Number of email addresses.	0	-
/UPnP/PHONE/AddressBook/PCC/Telecommunications/EMail/ Address/#/	MultiInstance	W	R	This MultiInstance contains the e-mail addresses component of the contact.	0	R

Name	Type	Acc.	Req	Description	EOC	Ver
Type	string	W	R	Values for this parameter are (the quotes shall not be used when specifying the values for this parameter): "Home", the address is for the whole home/family use "Business", the address is used for business "Personal", the address is for private uses "Other", any other purposes fro this address The value of this parameter shall be at least (see below) one of the above types, and shall NOT be the empty string. An attempt to set invalid values will result in the proper fault code returned by the device (CMS::705 "Invalid Value"). The Type is not required to be unique among the Instances. For example, in case there are two different business e-mail addresses, it is possible to have two different Instances having "Business" for Type value.	0	R
Address	string(256)	W	R	The internet e-mail address. [From [8]]	0	R
DisplayAs	string(64)	W	A	The formatted name used to display the address. For example, if the e-mail address is related to work, it might be desirable to display it as "work address". This parameter is an Outlook extension to vCard specification [8].	0	-
/UPnP/PHONE/AddressBook/PCC/Telecommunications/Web/	*SingleInstance*	-	A	Specifies the Web pages associated with the contact.	-	-
Preferred	unsignedInt	W	R	The *Instance* identifier of the preferred URL in the following list of Web Pages. This shall be an existing value. • When there are NO Instances of Pages, the value is 0 and can not be changed. An attempt to change that value shall return the proper fault code (CMS::705 "Invalid Value"). • When there is only one *Instance* of Pages, the value is the *Instance* identifier of such *Instance*. An attempt to set invalid values will result in the proper fault code returned by the device (CMS::705 "Invalid Value"). When there are more than a single *Instance* of Pages, it's up to the implementation to define the default value. An attempt to set invalid values will result in the proper fault code returned by the device (CMS::705 "Invalid Value"). For example, if there are the *Instance* identifier 3, 5 and 67, the control point can set this parameter to 3, 5 or 67. Any other values will produce an error.	0	-
PageNumberOfEntries	unsignedInt	-	R	Number of web pages.	0	-
/UPnP/PHONE/AddressBook/PCC/#/Telecommunications/Web /Page/#/	*MultiInstance*	W	R	This *MultiInstance* contains the Web Pages component of the contact.	-	-

Name	Type	Acc.	Req	Description	EOC	Ver
Type	string	W	R	Values for this parameter are (the quotes shall not be used when specifying the values for this parameter): "Home", the address is for the whole home/family use "Business", the address is used for business "Personal", the address is for private uses "Other", any other purposes fro this address The value of this parameter shall be at least (see below) one of the above types, and shall NOT be the empty string. An attempt to set invalid values will result in the proper fault code returned by the device (CMS::705 "Invalid Value"). The Type is not required to be unique among the Instances. For example, in case there are two different business web pages, it is possible to have two different Instances having "Business" for Type value.	0	-
URL	string(256)	W	A	The Web address. It shall be a valid URL.	0	-
/UPnP/PHONE/AddressBook/PCC/#/Geographical/	*SingleInstance*	-	A	This *SingleInstance* is concerned with information associated with geographical positions or regions associated with the contact	-	-
TZ	string(64)	W	A	Specifies information related to the time zone of the contact. The default is a single UTC-offset value. It can also be reset to a single text value. For example: "05:00; EST; Raleigh/North America" [From [8]]	0	-
Geo	string(64)	W	A	To specify information related to the global positioning of the contact. This type specifies information related to the global position of the contact. The value specifies latitude and longitude, in that order (i.e., "LAT LON" ordering). The longitude represents the location east and west of the prime meridian as a positive or negative real number, respectively. The latitude represents the location north and south of the equator as a positive or negative real number, respectively. The longitude and latitude values shall be specified as decimal degrees and should be specified to six decimal places. This will allow for granularity within a meter of the geographical position. The string components are separated by the SEMI-COLON character (ASCII decimal 59). The simple formula for converting degrees-minutes-seconds into decimal degrees is: decimal = degrees + minutes/60 + seconds/3600. This is an example of Geo value: "37.386013;-122.082932". [From [8]]	0	-

Name	Type	Acc.	Req	Description	EOC	Ver
/UPnP/PHONE/AddressBook/PCC/Organizational/	*SingleInstance*	-	A	This *SingleInstance* is concerned with information associated with characteristics of the organization or organizational units of the contact.	-	-
Title	string(64)	W	R	The job title, functional position or function of the contact. For example: "Director\, Research and Development" Note that the COMMA character (ASCII decimal 44) is escaped with the back-slash \. [From [8]]	0	-
Role	string(64)	W	R	Specifies information concerning the role, occupation, or business category of the contact. This parameter is included as an organizational type to avoid confusion with the semantics of the Title parameter and incorrect usage of that type when the semantics of this parameter is intended. For example: "Programmer" [From [8]]	0	-
Agent	string(64)	W	R	Specifies information about another person who will act on behalf of the individual or resource associated with the contact. A key characteristic of the Agent type is that it represents somebody or something that is separately addressable. Possible, and mutually exclusive value types for this parameter are: • A single vCard value (RFC 2426). For example "BEGIN:VCARD\nFN: …END:VCARD\n". • A reference to a contact in this AddressBook. For Example: "/UPnP/PHONE/AddressBook/Contact/45/". This extends the to vCard specification [8]. • An URI to specify information outside of this scope. For example: CID:JQPUBLIC.part3.960129T083020.xyzMail@host3.com". "URI: [From [8]]	0	-
Org	string(64)	W	R	Specifies the organizational name and units. This is a string consisting of components separated the SEMI-COLON character (ASCII decimal 59). The first part is the organization name, followed by one or more levels of organizational unit names. For example:"ABC\, Inc.;North American Division;Marketing". [From [8]]	0	-
/UPnP/PHONE/AddressBook/PCC/Organizational/Logo/	*SingleInstance*	-	A	Graphic image of a logo associated with the contact.	-	-

Name	Type	Acc.	Req	Description	EOC	Ver
MIMEType	string	W	R	The type of the following Value parameter. Allowed values, as a subset from the IANA registered formats http://www.iana.org/assignments/media-types/, are: • text/plain (http://www.iana.org/assignments/media-types/text/). • all image/… media types (http://www.iana.org/assignments/media-types/image/). [From [8]]	0	-
Value	string(32k)	W	R	The "value" of the logo, depending on the MIMEType: • If the MIMEType is text/plain this parameter shall contain the URI where the contact's logo is defined. For example: http://www.abc.com/pub/logos/jqpublic.gif • If the MIMEType is image/… this parameter shall contain the logo encoded with respect to the image MIME type specified. For example, if the MIMEType is image/jpg the Value shall contain the encoded jpg logo image. [From [8]]	0	-
/UPnP/PHONE/AddressBook/PCC/Explanatory/	*SingleInstance*	-	A	This *SingleInstance* contains additional explanations, such as that related to informational notes or revisions specific to the contact.	-	-
Categories	string	W	R	Specifies application category information about the contact description (e.g. the vCard). One or more text values separated by a COMMA character (ASCII decimal 44). For example: "TRAVEL AGENT" or "INTERNET,IETF,INDUSTRY,INFORMATION TECHNOLOGY" Are valid values for this parameter. [From [8]]	0	-
Note	string(256)	W	R	Specifies supplemental information or a comment that is associated with the contact description (e.g. the vCard). For example: "This fax number is operational 0800 to 1715 EST, Mon-Fri". [From [8]]	0	-
Prod	string(64)	W	A	Specifies the identifier for the product that created the contact information (e.g. the vCard). Implementations should use a method such as that specified for Formal Public Identifiers in ISO 9070 to assure that the text value is unique. For example: "-//ONLINE DIRECTORY//NONSGML Version 1//EN". [From [8]]	0	-

Name	*Type*	*Acc.*	Req	Description	*EOC*	*Ver*
Rev	string(64)	W	A	Specifies revision information about the contact description (e.g. the vCard). The value distinguishes the current revision of the information in this AddressBook's contact for other renditions of the information. For example: "1995-10-31T22:27:10Z". [From [8]]	0	-

Name	_Type_	_Acc._	Req	Description	_EOC_	_Ver_
SortString	string(64)	W	R	Specifies the family name or given name text to be used for national-language-specific sorting of the FormattedName and Name within /UPnP/PHONE/AddressBook/Contact/#/Identification/. The sort string is used to provide family name or given name text that is to be used in locale- or national-language-specific sorting of the formatted name and structured name types. Without this information, sorting algorithms could incorrectly sort this contact within a sequence of sorted contacts in the AddressBook. When this parameter value is assigned to a contact, then this family name or given name value is used for sorting among contacts. For example, in the case of family name sorting, the following examples define common sort string usage with the FormattedName and Name parameters. /.../FormattedName: Rene van der Harten /../Name/FamilyName: van der Harten; /../Name/GivenName:Rene ... /../SortString: Harten /.../FormattedName:Robert Pau Shou Chang /.../Name/FamilyName:Pau /.../Name/GivenName: Shou Chang ... /.../SortString: Pau /.../FormattedName:Osamu Koura /.../Name/FamilyName:Koura /.../Name/GivenName:Osamu ... /.../SortString: Koura [From [8]]	0	-

Name	Type	Acc.	Req	Description	EOC	Ver
UID	string(256)	W	R	Specifies a value that represents a globally unique identifier corresponding to the individual or resource associated with the contact. This parameter value should be an IANA registered identifier format, but a non-standard format is anyway permitted. Example: "19950401-080045-40000F192713-0052". [From [8]]	0	-
URL	string(256)	W	R	Specifies a uniform resource locator associated with this contact. [From [8]]	0	-
/UPnP/PHONE/AddressBook/PCC/Explanatory/Sound/	SingleInstance	-	A	This SingleInstance is used to specify a digital sound content information that annotates some aspect of the contact. By default this type is used to specify the proper pronunciation of the name type value of contact.	-	-
MIMEType	string	W	R	The type of the following Value parameter. Allowed values, as a subset from the IANA registered formats http://www.iana.org/assignments/media-types/, are: • text/plain (http://www.iana.org/assignments/media-types/text/). • all audio/… media types (http://www.iana.org/assignments/media-types/audio/). [From [8]]	0	-
Value	string(32k)	W	R	The "value" of the sound, depending on the MIMEType: • If the MIMEType is text/plain this parameter shall contain the URI where the contact's sound is defined. For example: http://www.abc.com/pub/thiscontact/pronunciation.mpeg • If the MIMEType is audio/… this parameter shall contain the sound encoded with respect to the aoudio MIME type specified. For example, if the MIMEType is audio/mp4 the Value shall contain the encoded mp4 audio stream. [From [8]]	0	-
/UPnP/PHONE/Settings/MultiMedia/	SingleInstance		A	Multi Media folder containing images and sounds.	-	-
SoundNumberOfEntries	unsignedInt	-	R	Number of multimedia sounds.	-	-

43

Name	Type	Acc.	Req	Description	EOC	Ver
/UPnP/PHONE/Settings/MultiMedia/Sound/#/	MultiInstance	W	R	Folder for sounds and melodies. Device implementations, accordingly with the Access attribute definition in [7], MAY provide some restrictions as follows: • The device might provide default Instances for this Node: an attempt to delete this node might result in an error returned by the device. The device might also return an error when the control point attempts to change parameter values that are part of the factory default. • The device might deny the attempt to create a new Instance. In this case, the Access attribute value for this Instance shall be ReadOnly instead of ReadWrite. See Access attribute in [7].	-	-
Name	string(64)	W	R	Name of the sound/melody. This name is not required to be unique and helps the user to identify and recognize the desired sound/melody file.	-	-
MIMEType	string	W	R	The type of the following Value parameter. Allowed values, as a subset from the IANA registered formats http://www.iana.org/assignments/media-types/, are: • text/plain (http://www.iana.org/assignments/media-types/text/). • all audio/... media types (http://www.iana.org/assignments/media-types/audio/).	-	-
Value	string(32k)	W	R	The "value" of the sound, depending on the MIMEType: • If the MIMEType is text/plain this parameter shall contain the URI where the melody is defined. For example: http://www.abc.com/pub/thiscontact/pronunciation.mpeg • If the MIMEType is audio/... this parameter shall contain the melody encoded with respect to the audio MIME type specified. For example, if the MIMEType is audio/mp4 the Value shall contain the encoded mp4 audio stream.	-	-
UserDefined	boolean	R	A	This allowed parameter, when supported, can be used to distinguish among the user defined sounds and the factory default sounds.	-	-
/UPnP/PHONE/Settings/RingingProfiles/	SingleInstance	-	A	This Node contains parameters regarding profiles. Each profile is used to configure how the telephone will inform the user of some events like an incoming phone call or a message. There can be just one active profile at the same time and the control point can enable a specific profile to be the active one.	-	-

Name	Type	Acc.	Req	Description	EOC	Ver
Active	unsignedInt	-	R	The *Instance* identifier of the active profile from the list. This parameter ease the control point to identify the active profile without the need to read them all. At least one profile shall exist in each implementation. In case there is just one profile it shall be active. Refer to /UPnP/PHONE/ Settings/RingingProfiles/Profile/#/Active parameter for further details on how to choose the existing profile to be active.	0	R
Test	boolean	W	A	This parameter is used to test the ringing tone of the currently active profile. When the Test parameter is set to true (1) the device shall play the incoming calls' alert tone of the currently active profile. The duration of the play is vendor dependent. The value read from the TelCP is 1 while the tone is ringing. When the Test parameter is set to false (0) the device shall immediately stop playing the ringing tone. The value read from the TelCP is 0 while the tone is not ringing.	-	-
CallAlertTypeNumberOfEntries	unsignedInt	-	R	Number of call alert types.	-	-
MessageAlertTypeNumberOfEntries	unsignedInt	-	R	Number of message alert types.	-	-
ProfileNumberOfEntries	unsignedInt	-	R	Number of profiles.	-	-
/UPnP/PHONE/Settings/RingingProfiles/CallAlertType/#/	MultiInstance	-	R	This read only table contains all the possible calls alert types for the device. Refer to the profile settings parameters for further details of their usage.	-	-
Type	string	-	R	The call alert type supported by the device implementation. Examples of typical alert types are: "Melody", "Increasing Melody", "Vibration", "Vibration then Melody", and so on.	-	-
/UPnP/PHONE/Settings/RingingProfiles/MessageAlertType/#/	MultiInstance	-	R	This read only table contains all the possible messages alert types for the device. Refer to the profile settings parameters for further details of their usage.	-	-
Type	string	-	R	The message alert type supported by the device implementation. Examples of typical alert types are: "Melody", "Vibration", "Mute", and so on.	-	-

Name	Type	Acc.	Req	Description	EOC	Ver
/UPnP/PHONE/Settings/RingingProfiles/Profile/#/	*MultiInstance*	W	R	Table representing the PHONE profiles. Device implementations, accordingly with the Access attribute definition in CMS, MAY provide some restrictions to profiles as follows: • The device might provide default Instances for this Node: an attempt to delete this node might result in an error returned by the device. The device might also return an error when the control point attempts to change parameter values that are part of the factory default. • The device might deny the attempt to create a new Instance. In this case, the Access attribute value for this Instance shall be ReadOnly instead of ReadWrite. See Access attribute in CMS. There shall be at least one (default) profile in this table. As the control point changes some settings (e.g. the tone of the incoming calls, as well as the volume) the device might announce the event to the user by, for example, playing the selected tone or providing a short demo of the new desired volume.	-	-
Name	string	W	R	The name of the profile. Possible values for this profile name might be (the quotes shall not be used when specifying the values for this parameter): "Normal", "Silent", "Party", "Offline". Device implementations MAY provide their own list of profiles and the user might be allowed to change them or to add new ones, providing new names. For example, the control point might add the profile name "My Personal Profile".	-	-

Name	Type	Acc.	Req	Description	EOC	Ver
Active	boolean	W	R	This indicates which is the active profile. The following rules shall be obey by all device implementations: • There shall be exactly one active profile at the same time. Its Active value is 1. • As the selected profile becomes active, all the remaining ones shall be set to 0 (not active). • As an active profile is deleted (when permitted), the implementation will choose the new active profile among the other ones. As the Active profile changes, the telephone shall update the value of /UPnP/PHONE/ Settings/RingingProfiles/Active parameter. As the control point changes the active profile, the device might announce the event to the user by playing the incoming calls' alert tone.	-	-
/UPnP/PHONE/Settings/RingingProfiles/Profile/#/IncomingCalls/	SingleInstance	-	R	Contains the basic setting for the incoming calls of this profile.	-	-
CallAlertType	unsignedInt	W	R	Instance identifier representing the type of the alert. The types supported by the device shall be declared in the table /UPnP/PHONE/ Settings/RingingProfiles/CallAlertTypes/#/. The control point shall use supported types only. An attempt to set a non supported value should result in an error code (CMS::705 "Invalid Value") returned by the device.	0	R
RingTone	unsignedInt	W	R	The Instance identifier of the /UPnP/PHONE/ Settings/MultiMedia/Sounds/#/, where the melody is took from. The control point shall use supported tones only. An attempt to set a non supported value should result in an error code (CMS::705 "Invalid Value") returned by the device. As the control point changes the ring tone of the current profile, the device might announce the event to the user by playing the new tone.	-	-
Volume	unsignedInt	W	R	A number specifying the percentage of the maximum volume for the device. Allowed values are: • From 1 to 10, for volume levels from 10% to 100% of the maximum. • 0, from the mute option.	0	R
/UPnP/PHONE/Settings/RingingProfiles/Profile/#/Messages/	SingleInstance	-	R	Contains the basic setting for the incoming calls of this profile.	-	-

Name	Type	Acc.	Req	Description	EOC	Ver
MessageAlertType	unsignedInt	W	R	*Instance* identifier representing the type of the alert. The types supported by the device shall be declared in the table /UPnP/PHONE/Settings/RingingProfiles/MessageAlertTypes/#/. The control point shall use supported types only. An attempt to set a non supported value should result in an error code (CMS::705 "Invalid Value") returned by the device.	0	R
MessageTone	unsignedInt	W	R	The *Instance* identifier of the /UPnP/PHONE/Settings/MultiMedia/Sounds/#/, where the melody is took from. The control point shall use supported melodies only. An attempt to set a non supported value should result in an error code (CMS::705 "Invalid Value") returned by the device. As the control point changes the ring tone of the current profile, the device might announce the event to the user by playing the new tone.	-	-
AlertRepetition	unsignedInt	W	R	The time lapse from an alert due to an incoming message to the next alert for the same unhandled message. The time shall be expressed in minutes and it's up to the device implementation to define whether to impose a maximum value. As the AlertRepetition value is 0, no repetition of the alert will be performed.	-	-
Volume				A number specifying the percentage of the maximum volume for the device. Allowed values are: • From 1 to 10, for volume levels from 10% to 100% of the maximum. • 0, from the mute option.	0	R
/UPnP/PHONE/Settings/AnsweringMachine/	*SingleInstance*	-	A	Configuration of the answering machine. Messages are recorded using some mean out of this specification scope and then saved. It is up to the device implementation to show the list of available voice messages in the *MultiInstance* node /UPnP/PHONE/AnsweringMachine/VoiceMessage/#/. The control point can select the desired voice message and choose whether it shall be active or not.	-	-
Enable	boolean	W	R	Configure whereas the answering machine is enabled or not. The default value for this parameter is 0 (not enabled).	0	R
SelectedVoiceMessage	unsignedInt	W	R	*Instance* identifier representing the selected VoiceMessage for the answering machine. The voice messages shall exist in the table /UPnP/PHONE/AnsweringMachine/VoiceMessage/#/. The control point shall use existing values only. An attempt to set a non existing Instance value should result in an error code (CMS::705 "Invalid Value") returned by the device.	0	R
VoiceMessageNumberOfEntries	unsignedInt	-	R	Number of voice messages	0	R

Name	Type	Acc.	Req	Description	EOC	Ver
/UPnP/PHONE/Settings/AnsweringMachine/VoiceMessage/#/	*MultiInstance*	W	R	Folder for recorded voice messages. Device implementations, accordingly with the Access attribute definition in [7], MAY provide some restrictions as follows: • The device might provide Instances for this Node: an attempt to delete this node might result in an error returned by the device. The device might also return an error when the control point attempts to change parameter values that are part of the factory default. • The device might deny the attempt to create a new Instance. In this case, the Access attribute value for this Instance shall be ReadOnly instead of ReadWrite. See Access attribute in [7].	-	-
Name	string(64)	W	R	Name of the recorded voice message. This name is not required to be unique and helps the user to identify and recognize the desired voice message file.	-	-
MIMEType	string	W	R	The type of the following Value parameter. Allowed values, as a subset from the IANA registered formats http://www.iana.org/assignments/media-types/, are: • text/plain (http://www.iana.org/assignments/media-types/text/). • all audio/... media types (http://www.iana.org/assignments/media-types/audio/).	-	-
Value	string(32k)	W	R	The "value" of the voice message, depending on the MIMEType: • If the MIMEType is text/plain this parameter shall contain the URI where the voice message is defined. For example: http://www.abc.com/pub/thiscontact/message.mpeg • If the MIMEType is audio/... this parameter shall contain the message encoded with respect to the audio MIME type specified. For example, if the MIMEType is audio/mp4 the Value shall contain the encoded mp4 audio stream.	-	-
/UPnP/PHONE/Settings/GPSInfo/	*SingleInstance*	-	A	Provides geographic coordinates for reading GPS data of a telephone equipped with a GPS receiver.	-	-

Name	Type	Acc.	Req	Description	EOC	Ver
Latitude	int	-	R	Latitude position as defined in http://en.wikipedia.org/wiki/Latitude. The latitude is represented in decimal degrees (see: http://en.wikipedia.org/wiki/Decimal_degrees) having 5 fixed decimal places accuracy (1.11 m). To encode the value in the SOAP "int" standard type, it shall be multiplied by 100000. For example, the decimal degree representation of the latitude location of the United States Capitol is 38.88972, when using 5 decimal places accuracy, therefore 3888972 is the value for the Latitude parameter.	-	-
Longitude	int	-	R	Longitude position as defined in http://en.wikipedia.org/wiki/Longitude. The latitude is represented in decimal degrees (see: http://en.wikipedia.org/wiki/Decimal_degrees) having 5 fixed decimal places accuracy (1.11 m). To encode the value in the SOAP "int" standard type, it shall be multiplied by 100000. For example, the decimal degree representation of the latitude location of the United States Capitol is -77.00888, when using 5 decimal places accuracy, therefore -7700888 is the value for the Longitude parameter.	-	-
Height	int	-	R	Height, expressed in meters.	-	-
/UPnP/PHONE/Settings/Power/	SingleInstance	-	A	Provides information regarding power source.	-	-
CurrentPowerSource	string	-		The current power source type. Possible values are: "AC Power" "Battery"	1	R
/UPnP/PHONE/Settings/Power/Battery/	SingleInstance	-	A	Provides information regarding the battery.	-	-
Status	String	-		The current status of the battery. Possible values are: "Available" means battery installed and working "Charging" means battery installed and under charging "Unavailable" means battery not installed "Error" means battery installed but malfunctioning	-	-
CurrentPowerLevel	int	-		The current power level of the battery, as percent value (0 means battery completely uncharged or battery not installed; 100 means battery fully charged).	-	-
LowBatteryAlarmLevel	int	W		The configurable threshold level of alarm for low battery, as a percent value.	1	R
LowBatteryAlarm	boolean	-		When true (1) indicates that the CurrentPowerLevel is less than LowBatteryAlarmLevel.	1	R

Name	Type	Acc.	Req	EOC	Ver	Description
/UPnP/PHONE/UserPreferences/	*SingleInstance*	-	A	-	-	Stores user preference settings for the WAN side telephony services settings and UPnP Telephony related service (e.g., Messaging and CallManagement) settings.
/UPnP/PHONE/UserPreferences/ServiceSettings/	*SingleInstance*	-	A	-	-	Stores user preference settings for the WAN side telephony services.
ActiveProfile	String	W	A	0	-	Stores current active profile name to be applied for the WAN side telephony services . e.g., "Home" or "Busy".
/UPnP/PHONE/UserPreferences/ServiceSettings/Settings/#/	*MultiInstance*	W	R	-	-	Store actual settings applicable for the WAN side messaging and call services.
ProfileName	*String*	W	A	0	-	Stores the profile name for the particular settings. For e.g. "Busy".
/UPnP/PHONE/UserPreferences/ServiceSettings/Settings/#/MessagingServiceSettings/	*SingleInstance*	-	R	-	-	Stores WAN side messaging settings. The implementation should implement at least this object or the CallServiceSettings object.
NetworkStorage	boolean	W	A	0	-	Enable/disable network storage for the messaging services.
DeferDelivery	boolean	W	A	0	-	Incoming message handling related settings. When message notification is received by the Telephony server from the WAN Side, the message can be either deferred or retrieved immediately. This setting controls the behavior of the Telephony Server. When this element is set to true then message will be deferred else it will be immediately retrieved by the TS.
SyncNetworkStorage	boolean	W	A	0	-	Sync local messages with network storage (especially for the messages like MMS and Email).
MessageMediaTypes	String	W	A	0	-	This identifies allowed types of the message to received or sent. Allowed values, as a subset from the IANA registered formats http://www.iana.org/assignments/media-types/ , are: • text/plain (http://www.iana.org/assignments/media-types/text/). • all audio/... media types (http://www.iana.org/assignments/media-types/audio/).
ConversationHistory	boolean	W	A	0	-	Enable/Disable conversation history storage for e.g. storing all the chat sessions.

Name	Type	Acc.	Req	Description	EOC	Ver
DisableServiceFeatures	String	W	A	Comma separated feature list to be disable for the messaging service. The allowed values for the features are as follows, • File Transfer • Group Chat • MMS • SMS • Email • Chat Implementation can add more service specific features. For example, "FileTransfer, GroupChat" will disable the corresponding features.	0	-
EnhancedOutgoingSession	String	W	R	This setting indicates additional information to be included in the outgoing chat sessions or message. The following are the allowed values for the additional information: • Image: to include image into the outgoing chat session or message. • location: to include the location information. Implementation can add more service specific elements. This element can carry multiple values separated by comma e.g., "image,location". The empty string (" ") will disable any kind of additional information from the outgoing message.	0	-
/UPnP/PHONE/UserPreferences/ServiceSettings/ServiceSettings/Settings/#/MessagingServiceSettings/AutoReply	SingleInstance	-	A	Small Auto reply feature for messaging services like email or SMS.	-	-
Status	boolean	W	R	Enable/Disable the auto reply message .	0	-
ServiceType	String	W	A	This represents the service name in which the auto reply is applicable, The allowed values for the services are: • SMS • Email Implementation can add more service specific elements. This element can carry multiple values separated by comma e.g., "SMS,Email".	0	-
MessageText	String	W	R	Text message to be set as a auto reply.	0	-

Name	Type	Acc.	Req	Description	EOC	Ver
/UPnP/PHONE/UserPreferences/ServiceSettings/Settings/#/MessagingServiceSettings/Report	*SingleInstance*	-	A	Enable disable the delivery and read report for the messages.	-	-
DeliveryReport	boolean	W	R	Enable/disable the delivery report for incoming messages.	0	-
ReadReport	boolean	W	R	Enable/disable the read report for incoming messages.	0	-
/UPnP/PHONE/UserPreferences/ServiceSettings/Settings/#/CallServiceSettings/	*SingleInstance*	-	R	Stores service settings for telephony call service. The implementation should implement at least this object or the MessagingServiceSettings object	-	-
MediaCapabilities	*String*	W	A	This parameter represents the allowed media to be used for the telephony call service. The allowed values for the media are: • Audio • Video Implementation can add more service specific elements. This element can carry multiple values separated by comma e.g., "Audio,Video".	0	-
CallDivert	*Boolean*	W	R	Enable/Disable the incoming call diverting.	0	-
/UPnP/PHONE/UserPreferences/HomeSettings/	*SingleInstance*	-	A	Stores user preference settings for the UPnP Telephony related services settings.	-	-
ActiveProfile	String	W	A	Stores current active profile name to be applied for the UPnP Telephony service settings e.g., "Evening" or "Busy".	0	-
/UPnP/PHONE/UserPreferences/HomeSettings/PreferenceSettings/#/	*MultiInstance*	-	R	Store actual settings for the UPnP Telephony service settings.	-	-
ProfileName	*String*	W	A	Stores the profile name for the particular settings. For e.g. "Busy".	0	-
/UPnP/PHONE/UserPreferences/HomeSettings/PreferenceSettings/#/MessagingPreferenceSettings/	*SingleInstance*	-	R	Stores Telephony setting related to the UPnP Messaging service. If the TS implements the Messaging Service then this object should be implemented.	-	-
MessageNotificationFields	*String*	W	R	Comma separated message structure elements to be included in the NewMessage event notification of the *Messaging* service. E.g., "from,subject, priority" Elements refer to the Message structure in the Messaging service.	0	-

53

Name	Type	Acc.	Req	Description	EOC	Ver
DisabledMessageClasses	String	W	A	Comma separated message classes which are disabled in the UPnP Messaging service. E.g., "SMS,EMAIL,MMS,Chat". Elements refer to the Message Class in the Messaging service. The TelCP(s) cannot send or receive the disabled message class.	0	-
/UPnP/PHONE/UserPreferences/HomeSettings/PreferenceSett ings/#/CallManagementPreferenceSettings/	SingleInstance	-	A	Stores Telephony setting related to the UPnP CallManagement service.	-	-
/UPnP/PHONE/UserPreferences/HomeSettings/PreferenceSett ings/#/CallManagementPreferenceSettings/AnsweringMessage /	SingleInstance	-	R	Stores the answering message related settings for the UPnP CallManagement service	-	-
/UPnP/PHONE/UserPreferences/HomeSettings/PreferenceSett ings/#/CallManagementPreferenceSettings/Message/#/	MultiInstance	-	R	Represents the answering message. If both URI and CDSobject are not present or if the values are set as the empty string (""), then the CallManagement service may use local message.	-	-
URI	uri	W	A	This element represents the URI of the recorded message.	0	-
CDSobject	string	W	A	This element represents the message stored in a UPnP ContentDelivery Service. The format is "UUID,serviceId,ObjectID".	0	-
Caller	String	W	R	This element represents the contact of the caller. The allowed values for this element is as follows: • URI (e.g., Tel URI, SIP URI etc). • InstanceID, this is the instance id of the /UPnP/PHONE/AddressBook/Contact/ element • The empty string ("") indicates this message is applicable to all the caller. This element may contain comma separated values of URIs and InstanceIds (e.g., "Tel:+919817171717,12").	0	-
/UPnP/PHONE/ConnectedDevices/	SingleInstance	-	A	The element stores the information of the devices that are connected to the TS.	-	-
/UPnP/PHONE/ConnectedDevices/Device/#/	MultiInstance	W	R	The element stores the device information.	-	-

Name	Type	Acc.	Req	Description	EOC	Ver
DeviceType	String	W	R	Identifies type of device. The allowed values are: • Mobile • Tablet • TV • PC Implementation can add more specific elements.	-	-
/UPnP/PHONE/ConnectedDevices/Device/#/Hardware/	SingleInstance	-	A	This element stores hardware capabilities of the device.	-	-
AvailableMemory	String	W	A	Identifies if the device has memory storage capability. This represents the size as a string. For e.g: "None", "32GB"	-	-
Battery	Boolean	W	A	Identifies if device is using battery.	-	-
Microphone	Boolean	W	A	Identifies if device has the microphone functionality.	-	-
Speaker	Boolean	W	A	Identifies if device has the speaker functionality.	-	-
/UPnP/PHONE/ConnectedDevice/Device/#/Hardware/Display/	SingleInstance	-	A	This element stores display capabilities of the device.	-	-
DisplayType	String	W	A	This element represents the type of the display. The allowed values are: • Projector • LCD • LED Implementation can add more specific elements.	-	-
Resolution	String	W	A	This element represents the resolution of the screen e.g., "720p".	-	-
3DSupport	Boolean	W	A	This elements represents if the display supports 3D capabilities of not.	-	
/UPnP/PHONE/ConnectedDevices/Device/#/Hardware/Camera/	SingleInstance	-	A	This element stores camera capabilities of the device.	-	
Resolution	Integer	W	A	This element represents the resolution of the camera. The resolution is represented as an integer in size of mega pixel. E.g., 8, 10 etc.	-	
FlashSupport	Boolean	W	A	This represents if device is having flash capabilities or not.	-	
/UPnP/PHONE/ConnectedDevices/Device/#/Hardware/UserInterface/	SingleInstance	-	A	This element stores user interface supported by the device.	-	
Touch	Boolean	W	A	This identifies if device is touch capable.	-	
Key-Pad	Boolean	W	A	This identifies if device supports keypad.	-	

55

Name	Type	Acc.	Req	Description	EOC	Ver
KeyBoard	Boolean	W	A	This identifies if device supports full keyboard.	-	
PointingDevice	Boolean	W	A	This identifies if device supports any pointing device like mouse.	-	
/UPnP/PHONE/ConnectedDevices/Device/#/Software/	SingleInstance	-	A	This element stores hardware capabilities of the device.	-	
/UPnP/PHONE/ConnectedDevices/Device/#/Software/OS/	SingleInstance	-	A	Identifies the OS details of the device.	-	
OSname	String	W	A	Represents the name of the OS, like Android/Windows.	-	
OSversion	String	W	A	Version of the OS.	-	
Vendor	String	W	A	Represents the vendor information of the OS.	-	
/UPnP/PHONE/ConnectedDevices/Device/#/Software/Push/	SingleInstance	-	A	Identifies the OS details of the device.	-	
PushAppIDs	String	W	A	Identifies the push application supported by the device.	-	
/UPnP/PHONE/ConnectedDevices/Device/#/Software/MediaSupports/	SingleInstance	-	A	Identifies the Media supported by the device.	-	-
Audio	Boolean	W	A	Identifies if device can handle audio.	-	-
Video	Boolean	W	A	Identifies if device can handle video.	-	-
MIMETYPE	String	W	A	This allowed element will be used to represent the supported MIMETYPE by the device.	-	-

Annex B
(informative)

Theory of Operation

Annex B explains several scenarios to illustrate the use of CMS to manage the *Phone Data Model* supported by the TS. The TelCP can use the actions from CMS to browse, search, read or write the *Parameter* values, and to create or delete the rows (i.e. *Instance Nodes*) in the tables (i.e. *MultiInstance Nodes*).

Annex B is not intended to explain how the CMS works, therefore the knowledge of the [7] is required to properly understand the following examples.

B.1   Browsing the *Phone Data Model*

Using the CMS actions it is possible for the TelCP to browse the actual data model provided by the TS, in order to discover:

- The list of supported data model specifications, and
- The complete list of supported parameters.

The *GetSupportedDataModels()*, *GetSupportedParameters()* and *GetInstances()* actions can be used for the browsing the *Phone Data Model*.

B.2   Telephony Server Administration

The TelCP can read the status or perform the basic administrative tasks of the network interfaces (e.g., WAN interfaces like GSM, GPRS, UMTS, POTS, ISDN, etc, and LAN interfaces like WiFi, Ethernet, USB, Bluetooth, etc), or the other high level connectivity (e.g., VOIP connection) of the TS using the table `/UPnP/PHONE/Interface/#/`. The TS lists all the supported and the manageable interface in this table. The each row of this table is corresponds to an interface of the TS.

For the each listed TS interface, the TelCP can:

- Read the name of the interface (*Parameter* `Name`);
- Detect the type of the interface (*Parameter* `Type`);
- Get the current status of the interface (*Parameter* `Status`);
- Get the last successful start time of the interface (*Parameter* `StartTime`);
- Enable or disable the interface (*Parameter* `Enable`).

The *GetInstances()*, *GetValues()* and *SetValues()* CMS actions can be used to manage these settings.

B.3   Managing the *Address Book*

The *PhoneManagement* provides the set of functions for managing the *Address Book Parameters* defined in the table `/UPnP/PHONE/AddressBook/`.

B.3.1   Retrieving all Contacts

A TelCP can retrieve the whole set of contacts from the *Address Book* by calling the *GetValues()* action. This action takes a *Parameters* as an input argument which will identify the set of requested *Parameters* or a table name (i.e. a *MultiInstance Node* in CMS terminology). In the case of retrieving all the contacts from the *Address Book*, the input argument will identify the table name of the *Address Book* (i.e.: `/UPnP/PHONE/AddressBook/`).

The *GetValues()* action returns the *ParameterValueList* output argument which will be the list all the *Parameters* belonging to the table of the *Address Book*. The following example will clarify the use of the *GetValues()* action.

The TelCP invokes *GetValues()* with the *Parameters* argument as:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<cms:ContentPathList
 xmlns:cms="urn:schemas-upnp-org:dm:cms"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:schemas-upnp-org:dm:cms
                   http://www.upnp.org/schemas/dm/cms.xsd">
   <ContentPath>/UPnP/PHONE/AddressBook/Contact/</ContentPath>
</cms:ContentPathList>
```

The *GetValues()* returns the *ParameterValueList* output argument which will return all the contacts :

```xml
<?xml version="1.0" encoding="UTF-8"?>
<cms: ParameterValueList
 xmlns:cms="urn:schemas-upnp-org:dm:cms"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:schemas-upnp-org:dm:cms
                   http://www.upnp.org/schemas/dm/cms.xsd">
<Parameter>
   <ParameterPath>/UPnP/PHONE/AddressBook/Contact/3/Identification/FormattedName</ParameterPath>
   <Value>Mr. John Doe</Value>
</Parameter>
<Parameter>
   <ParameterPath>/UPnP/PHONE/AddressBook/Contact/3/Identification/NickName</ParameterPath>
   <Value>MJD</Value>
</Parameter>
[. . .]
<Parameter>
   <ParameterPath>/UPnP/PHONE/AddressBook/Contact/3/Explanatory/Sound/Value</ParameterPath>
   <Value>MIICajCCAdOgAwIBAgICBEUwD…iBTeXN0</Value>
</Parameter>
<Parameter>
   <ParameterPath>/UPnP/PHONE/AddressBook/Contact/25/Identification/FormattedName</ParameterPath>
   <Value>Jane Doe Jr.</Value>
</Parameter>
<Parameter>
   <ParameterPath>/UPnP/PHONE/AddressBook/Contact/25/Identification/NickName</ParameterPath>
   <Value>Jane</Value>
</Parameter>
[. . .]
<Parameter>
   <ParameterPath>/UPnP/PHONE/AddressBook/Contact/25/Explanatory/Sound/Value</ParameterPath>
   <Value>dzELMAkGA1UEBhMCVVMxLDA…qBgNVBAoTI05ldHNjYX</Value>
</Parameter>
</cms:ParameterValueList>
```

B.3.2   Search for a Specific Contact

A TelCP can search for a specific contact in the *Address Book* using the *GetSelectedValues()* action. This action takes two input arguments the *StartingNode* and the *Filter*. The *Filter* input argument identifies the condition and the required piece of information. This action returns the *ParameterValueList* output argument which will contain the list of all *Parameters*, associated with their values, that satisfy the condition identified by the input arguments.

The following example will clarify the use of the *GetSelectedValues()* action

The TelCP invokes *GetSelectedValues()* action to search for all information in the *Address Book* related to Mr. John Doe, whose well known nickname is MJD. The *StartingNode* input argument is set to value

/UPnP/PHONE/AddressBook/Contact/#/

and the *Filter* input argument is set to value

/UPnP/PHONE/AddressBook/Contact/#/Identification/NickName = "MJD"

It is possible that in the *Address Book* there could be:

* No contact with the desired nickname, or
* Only one contact with the desired nickname, or
* Many contacts with the desired nickname.

Therefore, the number of contact listed in the output argument depends on the *Address Book* content. The example of the response below shows the case where only one contact matches the required nickname.

```
<?xml version="1.0" encoding="UTF-8"?>
<cms: ParameterValueList
 xmlns:cms="urn:schemas-upnp-org:dm:cms"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:schemas-upnp-org:dm:cms
                     http://www.upnp.org/schemas/dm/cms.xsd">
<Parameter>
   <ParameterPath>/UPnP/PHONE/AddressBook/Contact/3/Identification/FormattedName</P
arameterPath>
   <Value>Mr. John Doe</Value>
</Parameter>
<Parameter>
   <ParameterPath>/UPnP/PHONE/AddressBook/Contact/3/Identification/NickName</Parame
terPath>
   <Value>MJD</Value>
</Parameter>
[. . .]
<Parameter>
   <ParameterPath>/UPnP/PHONE/AddressBook/Contact/3/Explanatory/Sound/Value</Parame
terPath>
   <Value>MIICajCCAdOgAwIBAgICBEUwD…iBTeXN0</Value>
</Parameter>
</cms:ParameterValueList>
```

B.3.3   Retrieving Contacts with Common Properties

A TelCP can search for a list of contacts having common properties For example, the international prefix. A TelCP can use the *GetSelectedValues()* action with a *StartingNode* and

a *Filter* as two input arguments. The *Filter* input argument identifies the condition representing the common properties requested. This action returns the *ParameterValueList* output argument which contains the list of all *Parameters* that satisfy the condition indentified by the *Filter* input argument.

The following example will clarify the use of the *GetSelectedValues()* action for retrieving the contacts with common properties.

In this example, the TelCP invokes *GetSelectedValues()* to search for the contacts in the *Address Book* who lives in the U.S.A.. The input argument *StartingNode* input argument is set to value:

`/UPnP/PHONE/AddressBook/Contact/#/`

and the input argument *Filter* is set to the value:

`/UPnP/PHONE/AddressBook/Contact/#/DeliveryAddressing/Address/#/Country = "U.S.A."`

This action returns *ParameterValueList* output argument with all *Address Book* information for the contacts who lives in U.S.A.

B.3.4   Retrieving a Specific Contact

A TelCP can retrieve a specific contact information when the *Instance* identifier for that contact is known. The *GetValues()* action is used for this purpose. The following example will clarify the use of the *GetValues()* action for retrieving the specific contact information.

In this example the contact to retrieve is identified by the *Instance* identifier value 25. The TelCP invokes *GetValues()* with the *Parameters* as an input argument with the value:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<cms:ContentPathList
 xmlns:cms="urn:schemas-upnp-org:dm:cms"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:schemas-upnp-org:dm:cms
                  http://www.upnp.org/schemas/dm/cms.xsd">
   <ContentPath>/UPnP/PHONE/AddressBook/Contact/25/</ContentPath>
</cms:ContentPathList>
```

The *GetValues()* action returns the *ParameterValueList* output argument which includes the contact information for the contact which has *Instance* identifier value 25, as the example below shows.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<cms: ParameterValueList
 xmlns:cms="urn:schemas-upnp-org:dm:cms"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:schemas-upnp-org:dm:cms
                  http://www.upnp.org/schemas/dm/cms.xsd">
<Parameter>
   <ParameterPath>/UPnP/PHONE/AddressBook/Contact/25/Identification/FormattedName</
ParameterPath>
   <Value>Jane Doe Jr.</Value>
</Parameter>
<Parameter>
   <ParameterPath>/UPnP/PHONE/AddressBook/Contact/25/Identification/NickName</Param
eterPath>
   <Value>Jane</Value>
</Parameter>
[. . .other parameters about the contact. . .]
<Parameter>
   <ParameterPath>/UPnP/PHONE/AddressBook/Contact/25/Explanatory/Sound/Value</Param
eterPath>
```

```
      <Value>dzELMAkGA1UEBhMCVVMxLDA…qBgNVBAoTI05ldHNjYX</Value>
</Parameter>
</cms:ParameterValueList>
```

B.3.5   Updating Details of a Contact

A TelCP can update the contact information of any contact by using the *Instance* identifier of that contact. The *SetValues()* action is used for updating the contact information. The example below clarifies the use of *SetValues()* action for updating the contact information.

In this example, the contact is identified by the *Instance identifier* value 25. The TelCP uses the *SetValues()* action and set the *ParameterValueList* input argument value as mentioned below to change the telephone prefix and the number for that contact.

```
<?xml version="1.0" encoding="UTF-8"?>
<cms: ParameterValueList
 xmlns:cms="urn:schemas-upnp-org:dm:cms"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:schemas-upnp-org:dm:cms
                     http://www.upnp.org/schemas/dm/cms.xsd">
<Parameter>
   <ParameterPath>/UPnP/PHONE/AddressBook/Contact/25/Telecommunications/Tel/Number/
3/Number</ParameterPath>
   <Value>222-345</Value>
</Parameter>
<Parameter>
   <ParameterPath>/UPnP/PHONE/AddressBook/Contact/25/Telecommunications/Tel/Number/
3/Prefix</ParameterPath>
   <Value>(207)</Value>
</Parameter>
</cms:ParameterValueList>
```

B.3.6   Creating a New Contact

A TelCP can create a new contact entry into the *Address Book.* A new contact can be created using two mechanism as discussed below.

a) A new contact entry can be added to the *Address Book* by involving the *CreateInstance()* action and by setting the *MultiInstanceName* input argument value as:

   /UPnP/PHONE/AddressBook/Contact/

   The *CreateInstance()* action will return the *InstanceIdentifier* output argument which includes the *Instance* identifier for the new contact entry, for example:

   /UPnP/PHONE/AddressBook/Contact/21/

   This means that the new contact is successfully created and it can be addressed using the *Instance* identifier value 21. The TelCP can then use *SetValues()* action to set the contact information for that contact.

   Similarly each *Address Book* contact can contain nested tables (i.e. *MultiInstance Nodes*) like, for example, the *Email Address*. The TelCP can then create a new *Instance Node* for such *MultiInstance Node* by using *CreateInstance()*   action with the proper *MultiInstanceName* input argument.

   For example, if TelCP needs to create a new e-mail addresses for the contact above (identified by the *Instance* value 21), the TelCP can send the *CreateInstance()* action with the input argument *MultiInstanceName* value set to:

   /UPnP/PHONE/AddressBook/Contact/21/Telecommunications/EMail/Address/

b) A new contact entry can be added to the *Address Book* by invoking the *CreateInstance()* and using the input argument *ChildrenInitialization*. The input argument *ChildrenInitialization* can be used to initialize the new contact entries with contact

information in the *CreateInstance()* action. An example value of the *ChildrenInitialization* input argument for the contact is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<cms:ParameterInitialValueList
    xmlns:cms="urn:schemas-upnp-org:dm:cms"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="urn:schemas-upnp-org:dm:cms
                  http://www.upnp.org/schemas/dm/cms.xsd">
<Node>

    <ParameterInitializationPath>Identification/Name/FamilyName</ParameterInitializa
tionPath>
       <Value>Doe</Value>
    </Node>
    <Node>

    <ParameterInitializationPath>Identification/Name/GivenName</ParameterInitializat
ionPath>
       <Value>John</Value>
    </Node>
    </cms:ParameterInitialValueList>
```

B.3.7   Deleting a Contact

A TelCP can delete a contact entry from the *Address Book* by invoking the *DeleteInstance()* action. The input argument *InstanceIdentifier* identifies the contact entry to be deleted in the *DeleteInstance()* action. For example, if the contact entry having *Instance* identifier value 21 needs to be removed from the *Address Book*, then the TelCP shall invoke the *DeleteInstance()* action with the input argument *InstanceIdentifier* set to value as:

/UPnP/PHONE/AddressBook/21/

B.3.8   Adding Information to an Existing Contact

A TelCP can add new information into an existing contact entry of the *Address Book* by invoking the *CreateInstance()* action. The input arguments the *MultiInstanceName* and the *ChildrenInitialization* includes the new information for the contact.

The example below shows the values of the *MultiInstanceName* and *ChildrenInitialization* input arguments for adding the new business telephone number for the contact named John Doe in the *Address Book. The Instance* identifier *for the contact is 21.*

MultiInstanceName:

/UPnP/PHONE/AddressBook/Contact/21/Telecommunications/Tel/Number/

And, *ChildrenInitialization:*

```
<?xml version="1.0" encoding="UTF-8"?>
<cms:ParameterInitialValueList
 xmlns:cms="urn:schemas-upnp-org:dm:cms"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:schemas-upnp-org:dm:cms
                  http://www.upnp.org/schemas/dm/cms.xsd">
<Node>
    <ParameterInitializationPath>Type</ParameterInitializationPath>
    <Value>Mobile(business)</Value>
</Node>
<Node>
    <ParameterInitializationPath>Number</ParameterInitializationPath>
    <Value>+1122334455</Value>
</Node>
</cms:ParameterInitialValueList>
```

B.3.9   Managing CMS Notifications for Changes in the *Address Book*

A TelCP can subscribe to the event notification for any changes in the *Address Book* for example the addition of new contact entry, the deletion of a contact entry etc. The *Parameters* in the *Address Book* are required to support the *EventOnChange* attribute. A TelCP shall set *EventOnChange* attribute value to 1 (true) in order to receive the event on any changes in the *Parameter* values. A TelCP can invoke the *SetAttributes()* action to set the value of the *EventOnChange* attribute. The *SetAttributes()* action, with an input argument *NodeAttributeValueList,* can be used to set the *EventOnChange* attribute.

The example below shows the value of the *NodeAttributeValueList* input argument, for setting the *EventOnChange* attribute of the *Parameter* /UPnP/PHONE/AddressBook/ContactNumberOfEntries to 1. The attribute value of this *Parameter* is set to 1 for getting the notification on any addition or deletion of a contact entry in the *Address Book*.

```
<?xml version="1.0" encoding="UTF-8"?>
<cms:NodeAttributeValueList
xmlns:cms="urn:schemas-upnp-org:dm:cms"
xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation="urn:schemas-upnp-org:dm:cms
 http://www.upnp.org/schemas/dm/cms.xsd">
<Node>
<NodeAttributePath>
    /UPnP/PHONE/AddressBook/ContactNumberOfEntries
  </NodeAttributePath>
  <EventOnChange>1</EventOnChange>
</Node>
</cms:NodeAttributeValueList>
```

Whenever there is an update in the number of contacts in the *Address Book*, the CMS generates the *ConfigurationUpdate* event to the TelCP. The TelCP can retrieve the updates on contact instances by calling the *GetInstances()* action with input argument *SearchDepth* set to 1 and the input argument *StartingNode* argument set to value:

/UPnP/PHONE/AddressBook/Contact/

The *GetInstances()* action returns the *Result* output argument. For example, if the *Address Book* contains the contacts identified by the *Instance* identifiers 3, 4 and 7, then the value of the *Result* output argument will be as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<cms:InstancePathList
   xmlns:cms="urn:schemas-upnp-org:dm:cms"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:schemas-upnp-org:dm:cms
   http://www.upnp.org/schemas/dm/cms.xsd">
   <InstancePath>
      /UPnP/PHONE/AddressBook/Contact/3/
   </InstancePath>
   <InstancePath>
      /UPnP/PHONE/AddressBook/Contact/4/
   </InstancePath>
   <InstancePath>
      /UPnP/PHONE/AddressBook/Contact/7/
   </InstancePath>
</cms:InstancePathList>
```

and the TelCP can check this list with its own local copy of the *Address Book*.

B.4   Managing the PHONE Settings

In order to configure settings related to the ringing modes, the *Phone Data Model* defines a set of *Parameters* under the following *Node*:

`/UPnP/PHONE/Settings/`

B.4.1   Ringing Profiles

A "ringing profile", or a "profile" is a set of configuration *Parameters* of the telephone which are related to the ringing modes and sounds.

The phone can have the predefined profiles and the user-created profiles. This list in the *Phone Data Model* corresponds to the *Node*:

`/UPnP/PHONE/Settings/RingingProfiles/Profile/#/`

All the profiles are grouped in the *Phone Data Model* under the *Node*:

`/UPnP/PHONE/Settings/RingingProfiles/`

Only one profile is active at a time on the telephone. This active profile can be indentified by the parameter `/UPnP/PHONE/Settings/RingingProfiles/Active`.

B.4.2   Changing the Active Profile

The TelCP can change the active profile by selecting the desired one among the existing profiles. All existing profiles can be retrieved by invoking the *GetValues()* action.

The TelCP invokes the *GetValues()* action with the value of the input argument *Parameters* set to as:

`/UPnP/PHONE/Settings/RingingProfiles/Profile/`

The *GetValues()* returns the output argument *ParameterValueList*, which will contain all the profile information. The example of the value of the output argument *ParameterValueList* is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<cms: ParameterValueList
 xmlns:cms="urn:schemas-upnp-org:dm:cms"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:schemas-upnp-org:dm:cms
                     http://www.upnp.org/schemas/dm/cms.xsd">
<Parameter>
   <ParameterPath>/UPnP/PHONE/Settings/RingingProfiles/Profile/3/Name</ParameterPat
h>
   <Value>Normal</Value>
</Parameter>
<Parameter>
   <ParameterPath>/UPnP/PHONE/Settings/RingingProfiles/Profile/3/Enable</ParameterP
ath>
   <Value>0</Value>
</Parameter>
[. . .]
<Parameter>
   <ParameterPath>/UPnP/PHONE/Settings/RingingProfiles/Profile/#/Messages/Volume</P
arameterPath>
   <Value>3</Value>
</Parameter>
</cms:ParameterValueList>
```

The TelCP might show all the retrieved profiles to the user. After the user has selected the desired profile to be set on the telephone, the TelCP can invoke the *SetValues()* action. For an example the TelCP can set the profile having the *Instance* identifier value as 5 by invoking the *SetValues()* action with value of the *ParameterValueList* input argument as:

```
<?xml version="1.0" encoding="UTF-8"?>
<cms: ParameterValueList
 xmlns:cms="urn:schemas-upnp-org:dm:cms"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:schemas-upnp-org:dm:cms
                     http://www.upnp.org/schemas/dm/cms.xsd">
<Parameter>
    <ParameterPath>/UPnP/PHONE/Settings/RingingProfiles/Profile/5/Enable</ParameterP
ath>
    <Value>1</Value>
</Parameter>
</cms:ParameterValueList>
```

As the profile identified by *Instance* identifier value 5 has become active profile, the telephone should inform the user about this change by playing the incoming call ring tone of the profile.

B.4.3 Creating a New Profile

The TelCP can create a new profile using *CreateInstance()* action for the *Node*.

/UPnP/PHONE/Settings/RingingProfiles/Profile/#/.

This creates an new profile for the telephone, then the TelCP can use the *SetValues()* action for setting the actual values of the *Parameters* for the profile.

B.4.4 Activating the Answering Machine

The TelCP can configure the settings related to the answering machine feature on the phone. The *Phone Data Model* defines the following *Parameters*:

- `/UPnP/PHONE/Settings/AnsweringMachine/Enable` which allows to turn on and off the feature.

- `/UPnP/PHONE/Settings/AnsweringMachine/SelectedVoiceMessage` which allows to select the particular voice message from the available prerecorded voice messages identified by the Node `/UPnP/PHONE/Settings/AnsweringMachine/VoiceMessage/#/`.

To activate the answering machine feature and to set the voice message identified by the *Instance* identifier value 5, the TelCP invokes the *SetValues()* action. The input argument *ParameterValueList* will contain the activation of answering machine feature and the requested voice message. The following example shows the value of the *ParameterValueList*:

```
<?xml version="1.0" encoding="UTF-8"?>
<cms: ParameterValueList
 xmlns:cms="urn:schemas-upnp-org:dm:cms"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:schemas-upnp-org:dm:cms
                     http://www.upnp.org/schemas/dm/cms.xsd">
<Parameter>
    <ParameterPath>/UPnP/PHONE/Settings/AnsweringMachine/Enable</ParameterPath>
    <Value>1</Value>
</Parameter>
<Parameter>
    <ParameterPath>/UPnP/PHONE/Settings/AnsweringMachine/SelectedVoiceMessage</Param
eterPath>
    <Value>5</Value>
</Parameter>
```

```
</cms:ParameterValueList>
```

B.4.5   Managing Voice Messages for the Answering Machine

In the *Phone Data Model,* the table `UPnP/PHONE/Settings/AnsweringMachine/VoiceMessage/#/` identifies the list of prerecorded voice messages. The TelCP can add and remove the prerecorded voice messages from the list.

The TelCP can add a new voice message by invoking the *CreateInstance()* action with value of *MultiInstanceName* as:

/UPnP/PHONE/ Settings/AnsweringMachine/VoiceMessage/#/

The *ChildrenInitialization* input argument is used to initialize the *Parameters* of the voice message. The example below shows the value of the input argument *ChildrenInitialization.*

```
<?xml version="1.0" encoding="UTF-8"?>
<cms:ParameterInitialValueList
 xmlns:cms="urn:schemas-upnp-org:dm:cms"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:schemas-upnp-org:dm:cms
                     http://www.upnp.org/schemas/dm/cms.xsd">
<Node>
    <ParameterInitializationPath>Name</ParameterInitializationPath>
    <Value>Currently busy</Value>
</Node>
<Node>
    <ParameterInitializationPath>MIMEType</ParameterInitializationPath>
    <Value>audio/mp3</Value>
</Node>
<Node>
    <ParameterInitializationPath>Value</ParameterInitializationPath>
    <Value>dzELMAkGA1UEBhMCVVMxLDA…qBgNVBAoTI05ldHNjYX</Value>
</Node>
</cms:ParameterInitialValueList>
```

The *CreateInstance()* action will return the *Instance* identifier for the new voice message for example:

/UPnP/PHONE/Settings/AnsweringMachine/VoiceMessage/3/

B.4.6   Deleting Voice Messages from the Answering Machine

The TelCP can delete the existing voice message from the list by invoking the *DeleteInstance()* action with input argument *InstanceIdentifier* set to the *Instance* identifier of the voice message to be deleted, for example:

/UPnP/PHONE/Settings/AnsweringMachine/VoiceMessage/5/

Note that the prerecorded voice message will be effectively deleted or not depends on the device implementation.

B.5   Managing the Power Settings

In order to inform the TelCP about the current power source and battery status of the phone, the *Phone Data Model has* defined the set of *Parameters* under the following *Node*:

`/UPnP/PHONE/Settings/Power/`

B.5.1 Power Source Info

The Phone Data Model has defined the *Parameter* `/UPnP/PHONE/Settings/Power/CurrentPowerSource` to specify the type of power source of the telephone e.g. "AC power" or "Battery". This information can be retrieved at any time by a TelCP by reading the corresponding *Parameter*.

If the TelCP subscribes to the *PhoneManagement* events, the TelCP can then receive the event notification for any changes in the power source. Such eventing can be enabled by setting the value of the *EventOnChange* attribute for the `CurrentPowerSource` *Parameter* to 1. The TelCP can invoke the *SetAttributes()* action with the following value for the input argument *NodeAttributeValueList*:

```
<?xml version="1.0" encoding="UTF-8"?>
<cms:NodeAttributeValueList
xmlns:cms="urn:schemas-upnp-org:dm:cms"
xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation="urn:schemas-upnp-org:dm:cms
 http://www.upnp.org/schemas/dm/cms.xsd">
<Node>
  <NodeAttributePath>
    /UPnP/PHONE/Settings/Power/CurrentPowerSource
  </NodeAttributePath>
  <EventOnChange>1</EventOnChange>
</Node>
</cms:NodeAttributeValueList>
```

Whenever the TS changes its power source for e.g. from "AC Power" to "Battery" the *ConfigurationUpdate* state variable will be evented the TelCP.

As the event received by the TelCP does not contain information about the change, the TelCP shall use the *GetValues()* or the *GetSelectedValues()* action to retrieve the change in the `CurrentPowerSource` *Parameter*.

B.5.2 Monitoring the Battery Level

The Phone Data Model provides the *Parameters* (`Status`, and `CurrentPowerLevel`) regarding the current status and power level of the battery of the telephone under the *Node* `/UPnP/PHONE/Settings/Power/Battery/.`

The TelCP can retrieve such information by invoking the *GetValues()* on the these *Parameters*.

The *Phone Data Model* provides a way to a TelCP to set a threshold level for the notification when the battery level goes under a set threshold. For example, the TelCP can invoke the *SetValues()* action with the value of the input argument *ParameterValueList* as below for setting the value of threshold as 30%:

```
<?xml version="1.0" encoding="UTF-8"?>
<cms: ParameterValueList
 xmlns:cms="urn:schemas-upnp-org:dm:cms"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:schemas-upnp-org:dm:cms
                http://www.upnp.org/schemas/dm/cms.xsd">
<Parameter>
  <ParameterPath>/UPnP/PHONE/Settings/Power/Battery/LowBatteryAlarmLevel
</ParameterPath>
  <Value>30</Value>
</Parameter>
```

```
</cms:ParameterValueList>
```

To receive some information when the value of `CurrentPowerLevel` goes below 30, the TelCP shall also properly set its value of *EventOnChange* attribute to 1, using the *SetAttributes()* action as explained in B.5.1.

B.5.3   Receiving and Verifying Alarms

If the TS supports the *Version* attribute for the *Parameters*

```
/UPnP/PHONE/Settings/Power/CurrentPowerSource
```

and

```
UPnP/PHONE/Settings/Power/Battery/LowBatteryAlarmLevel
```

and it also supports the *GetSelectedValues()* action, then the TelCP can easily know which *Parameter* has changed.

For example, supposing that TelCP receives the following value in a *ConfigurationUpdate* event:

```
"379,2007-10-24T05:41:00"
```

This indicates the change in either the current power source or the battery level. The first value `379` indicates the value of the *CurrentConfigurationVersion* state variable. Then TelCP can invoke the *GetSelectedValues()* action with the values for the input arguments *Filter* and *StartingNode* as:

Filter= "`Version <= 379`"

StartingNode="`/UPnP/PHONE/Settings/Power/`"

The TS will return the name and the value of the *Parameter* which has changed. If the *ConfigurationUpdate* event was due to the change in the current power source, then the TS will return the *Parameter*

`/UPnP/PHONE/Settings/Power/CurrentPowerSource` and its current value. Otherwise if the *ConfigurationUpdate* event was due to a low battery level, then the TS will return the *Parameter*

`UPnP/PHONE/Settings/Power/Battery/LowBatteryAlarmLevel` and its current value.

B.6   Managing the User Preferences

In order to manage the service experience provided by the UPnP Telephony, the *Phone Data Model* provides the set of user preferences. The user preferences are divided in two preference settings;

- Service Settings:
    - These are the set of service related settings which will control the service behavior in the WAN side. The two main services considered are messaging and call.
    - User can manage the service settings as a profile which will allow user to manage the service settings easily. A User can associate an unique profile name for service settings.

- Home Settings:

  - These are the set of preferences for service settings which are related to UPnP Telephony services. These settings are applicable to home network. These settings are grouped into two service categories: *Messaging* Service and *CallManagement* Service.

  - These home preferences are also grouped in to a profile to provide an ease in managing the UPnP Telephony service settings.

The *GetValues()* and *SetValues()* CMS actions can be used to manage these settings.

B.7   Network address book handling

PhoneManagement defines the address book elements in the data model to support the handling of network address book. *Phone Data Model* defines the *NetworkAddressBookId* element for each of the contact elements in the address book. This element will be used to specify the network address book identifier or source identifier where the particular contact belongs. The example network address book identifier can be GMAIL, FACEBOOK, or OMA-CAB etc.

- Updating the contact information in the network address book

- *Setvalues()* action can be used to update the contact information in the local address book and the request also include the *NetworkAddressBookId* element which will allow TS to carry the same changes into the corresponding network address book.

- Adding the contact into the network address book

- *SetValues()* action can be used to add the new contact into the local address book and the request also includes the *NetworkAddressBookId* element which will allow TS to add the new contact into the corresponding network address book as well.

- Searching the contact from the network address book

- *GetSelectedValues()* action can be used for searching the contacts. The request includes the filter for the network address book identifier, e.g. /UPnP/PHONE/AddressBook/Contact/#/ NetworkAddressBookId = "gmail"

- The TS will search the contact in local address book and also from the network address book and send the search results to the TelCP.

B.8   Personal Contact Card (PCC) information handling

A Personal Contact Card (PCC) defines a user's own profile information. *Phone Data Model* defines an element in the address book as "PCC". This element contains all personal information of the user. The same structure as represented by "Contact" element in the address book is used to represent the user profile information.

The *GetValues()* and *SetValues()* CMS actions can be used to manage PCC information for the user.

B.9   Connected device information handling

In a home network different types of devices are connected together. The information about the different connected devices can be exposed to the WAN side which enriches communication service experience of the users. The WAN side communication services can take advantage of the home device capabilities e.g., if a TS is connected with the HD capable TV, then peer can initiate the HD Call with the home user. *Phone Data Model* defines

elements to store the device information and capabilities information of the devices in the data model.

The *GetValues()* and *SetValues()* CMS actions can be used to manage these information element.

Annex C
(informative)

Bibliography

The following documents, in whole or in part, may be useful for understanding this document' but they are not essential 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.

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

ICS 35.200

Price based on 71 pages