

---

## **PhoneManagement:1**

**For UPnP Version 1.0**

**Status: Standardized DCP (SDCP)**

**Date: March 22, 2011**

**Document Version: 1.0**

**Service Template Version: 2.00**

---

This Standardized DCP has been adopted as a Standardized DCP by the Steering Committee of the UPnP Forum, pursuant to Section 2.1(c)(ii) of the UPnP Forum Membership Agreement. UPnP Forum Members have rights and licenses defined by Section 3 of the UPnP Forum Membership Agreement to use and reproduce the Standardized DCP in UPnP Compliant Devices. All such use is subject to all of the provisions of the UPnP Forum Membership Agreement.

THE UPNP FORUM TAKES NO POSITION AS TO WHETHER ANY INTELLECTUAL PROPERTY RIGHTS EXIST IN THE STANDARDIZED DCPS. THE STANDARDIZED DCPS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS". THE UPNP FORUM MAKES NO WARRANTIES, EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE STANDARDIZED DCPS, INCLUDING BUT NOT LIMITED TO ALL IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE, OF REASONABLE CARE OR WORKMANLIKE EFFORT, OR RESULTS OR OF LACK OF NEGLIGENCE.

Copyright © 2011 UPnP Forum. All Rights Reserved.

| <b>Authors<sup>1</sup></b> | <b>Company</b>      |
|----------------------------|---------------------|
| Alessandro De Vincentis    | Telecom Italia      |
| Enrico Grosso (Editor)     | Telecom Italia      |
| Jooyeol Lee                | Samsung Electronics |
| Jeyoung Maeng              | Samsung Electronics |
| Massimo Messori            | Telecom Italia      |
| Davide Moreo (Editor)      | Telecom Italia      |
| Yoshiki Nishikawa          | NTT                 |
| Mayuresh Patil             | Samsung Electronics |
| Mahfuzur Rahman            | Samsung Electronics |
| Yu Zhu                     | Huawei              |

<sup>1</sup> Note: The UPnP Forum in no way guarantees the accuracy or completeness of this author list and in no way implies any rights for or support from those members listed. This list is not the specifications' contributor list that is kept on the UPnP Forum's website.

# Contents

|   |           |
|---|-----------|
| Contents.....   | 3         |
| List of Tables.....   | 5         |
| List of Figures .....   | 6         |
| <b>1 Overview and Scope .....</b>   | <b>7</b>  |
| 1.1 Introduction .....  | 7         |
| 1.2 Notation.....   | 7         |
| 1.2.1 Data Types .....  | 8         |
| 1.3 Vendor-defined Extensions .....   | 8         |
| 1.4 References.....   | 8         |
| 1.4.1 Normative References.....   | 8         |
| 1.4.2 Informative References .....  | 9         |
| <b>2 <i>ConfigurationManagement</i> Service Profile (Normative).....</b>      | <b>10</b> |
| 2.1 Service Type .....  | 10        |
| 2.2 Terms and Abbreviations .....   | 10        |
| 2.3 <i>PhoneManagement</i> Architecture.....                                  | 10        |
| 2.3.1 Phone Management.....   | 11        |
| 2.3.2 <i>Phone Data Model</i> .....   | 11        |
| 2.4 State Variables .....   | 12        |
| 2.5 Actions .....   | 13        |
| 2.5.1 <i>GetSelectedValues()</i> .....  | 14        |
| 2.5.2 <i>SetValues()</i> .....  | 15        |
| 2.5.3 <i>CreateInstance()</i> .....   | 15        |
| 2.5.4 <i>DeleteInstance()</i> .....   | 15        |
| 2.5.5 <i>SetAttributes()</i> .....  | 15        |
| <b>3 Theory of Operation (Informative) .....</b>                              | <b>16</b> |
| 3.1 Browsing the <i>Phone Data Model</i> .....                                | 16        |
| 3.2 Telephony Server Administration .....                                     | 16        |
| 3.3 Managing the <i>Address Book</i> .....                                    | 16        |
| 3.3.1 Retrieving all Contacts.....  | 16        |
| 3.3.2 Search for a Specific Contact.....                                      | 18        |
| 3.3.3 Retrieving Contacts with Common Properties.....                         | 18        |
| 3.3.4 Retrieving a Specific Contact .....                                     | 19        |
| 3.3.5 Updating Details of a Contact.....                                      | 20        |
| 3.3.6 Creating a New Contact .....  | 20        |
| 3.3.7 Deleting a Contact.....   | 21        |
| 3.3.8 Adding Information to an Existing Contact .....                         | 21        |
| 3.3.9 Managing CMS Notifications for Changes in the <i>Address Book</i> ..... | 22        |
| 3.4 Managing the PHONE Settings .....   | 23        |
| 3.4.1 Ringing Profiles .....  | 23        |
| 3.4.2 Changing the Active Profile.....  | 23        |

|                    |   |           |
|--------------------|---|-----------|
| 3.4.3              | Creating a New Profile.....                             | 24        |
| 3.4.4              | Activating the Answering Machine .....                  | 24        |
| 3.4.5              | Managing Voice Messages for the Answering Machine ..... | 25        |
| 3.4.6              | Deleting Voice Messages from the Answering Machine..... | 26        |
| 3.5                | Managing the Power Settings.....                        | 26        |
| 3.5.1              | Power Source Info .....                                 | 26        |
| 3.5.2              | Monitoring the Battery Level.....                       | 27        |
| 3.5.3              | Receiving and Verifying Alarms .....                    | 27        |
| <b>4</b>           | <b>XML Service Description .....</b>                    | <b>28</b> |
| <b>Appendix A.</b> | <b><i>Phone Data Model (Normative)</i> .....</b>        | <b>29</b> |

## List of Tables

|            |                                    |    |
|------------|------------------------------------|----|
| Table 2-1: | Abbreviations.....                 | 10 |
| Table 2-2: | State Variables for Eventing ..... | 13 |
| Table 2-3: | Required Actions .....             | 14 |

## List of Figures

|  |    |
|--|----|
| Figure 1: <i>PhoneManagement</i> via CMS and <i>Phone Data Model</i> ..... | 11 |
| Figure 2: Overview of the <i>Phone Data Model</i> .....                    | 30 |

# 1 Overview and Scope

This document specifies the [ConfigurationManagement](#) Service Profile (see section: 2, [ConfigurationManagement](#) Service Profile (Normative)) herein referred as [PhoneManagement](#).

The [PhoneManagement](#) can be used for managing the configuration of the UPnP [TelephonyServer](#) device (i.e. a telephone), for e.g. 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.

The [PhoneManagement](#) is a profile of the [ConfigurationManagement](#) service as defined in [CMS] by the UPnP [DeviceManagement](#) WC for configuration management of specific features of a phone for e.g. Address Book, Phone settings etc. For this purpose, the [PhoneManagement](#) defines

- A profile of the CMS service in order to satisfy the requirements for [PhoneManagement](#) (see section: 2).
- The *Phone Data Model* to perform configuration management of the specific features of a phone (see Appendix A) by reusing the CMS data model (see: Appendix B, Common Objects, in [CMS]).

## 1.1 Introduction

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 document 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.

## 1.2 Notation

- In this document, features are described as Required, Recommended, or Optional as follows:

The key words “MUST,” “MUST NOT,” “REQUIRED,” “SHALL,” “SHALL NOT,” “SHOULD,” “SHOULD NOT,” “RECOMMENDED,” “MAY,” and “OPTIONAL” in this specification are to be interpreted as described in [RFC 2119].

In addition, the following keywords are used in this specification:

PROHIBITED – The definition or behavior is an absolute prohibition of this specification.  
Opposite of REQUIRED.

CONDITIONALLY REQUIRED – The definition or behavior depends on a condition. If the specified condition is met, then the definition or behavior is REQUIRED, otherwise it is PROHIBITED.

CONDITIONALLY OPTIONAL – The definition or behavior depends on a condition. If the specified condition is met, then the definition or behavior is OPTIONAL, otherwise it is PROHIBITED.

These keywords are thus capitalized when used to unambiguously specify requirements over protocol and application features and behavior that affect the interoperability and security of implementations. When these words are not capitalized, they are meant in their natural-language sense.

- 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 the UPnP Device Architecture are printed using the *arch* character style.
- A double colon delimiter, “::”, 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.

### 1.2.1 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 [DEVICE]. The XML Schema namespace is used to define property data types [XML SCHEMA-2].

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 MUST NOT be sent out by devices but MUST 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 MUST NOT be sent out by devices but MUST be accepted on input.

## 1.3 Vendor-defined Extensions

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

## 1.4 References

### 1.4.1 Normative References

This section lists the normative references used in this specification and includes the tag inside square brackets that is used for each such reference:

[DEVICE] – UPnP Device Architecture, version 1.0, UPnP Forum, June 13, 2000.

Available at: <http://www.upnp.org/specs/architecture/UPnP-DeviceArchitecture-v1.0-20000613.htm>.

Latest version available at: <http://www.upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.0.pdf>.

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

Available at: [ISO 8601:2000](http://www.iso.org/iso/iso_8601_2000).

[RFC 2119] – 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>.



[RFC 3339] – 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>.

[XML] – 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>.

[XML SCHEMA-2] – 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>.

[CMS] – *UPnP ConfigurationManagement:1 Service Document*, UPnP Forum, July 5, 2010. Available at: <http://www.upnp.org/specs/arch/UPnP-dm-ConfigurationManagement-v1.0-Service.pdf>

[VCARD] – 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>.

### 1.4.2 Informative References

This section lists the informative references that are provided as information in helping understand this specification:

[PHONEARCH] – *TelephonyArchitecture:1*, UPnP Forum, March 22, 2011.

Available at: <http://www.upnp.org/specs/phone/UPnP-phone-TelephonyArchitecture-v1-20110322.pdf>.

Latest version available at: <http://www.upnp.org/specs/phone/UPnP-phone-TelephonyArchitecture-v1.pdf>.

## 2 **ConfigurationManagement** Service Profile (Normative)

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

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

PhoneManagement is CONDITIONALLY REQUIRED if the *Phone Data Model* is supported by the Telephony Server.

### 2.1 Service Type

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

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

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

### 2.2 Terms and Abbreviations

Table 2-1: Abbreviations

| Abbreviation | Description                      |
|--------------|----------------------------------|
| CMS          | Configuration Management Service |
| TC           | Telephony Client                 |
| TelCP        | Telephony Control Point          |
| TS           | Telephony Server                 |

### 2.3 **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 inline with the Telephony architecture.

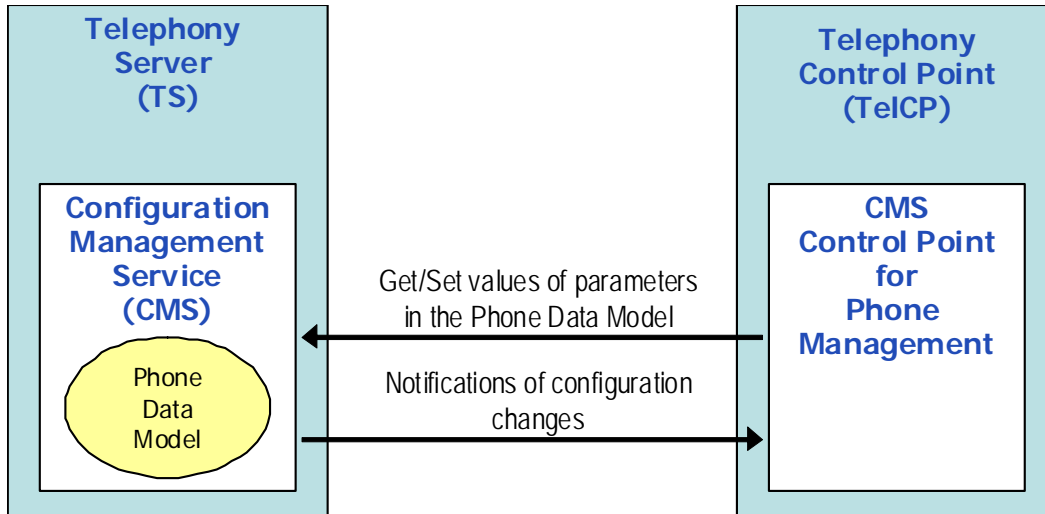


Figure 1: ***PhoneManagement*** via CMS and *Phone Data Model*

### 2.3.1 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 *Address Book*).
- 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 fulfill 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.

### 2.3.2 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 mandatory 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.

### 2.3.2.1 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 MUST be supported by all CMS implementations (see [CMS] 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 optional for the CMS, but these attribute MUST 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 must be generated). The *Phone Data Model* in Appendix A defines the list of *Parameters* which must support [EventOnChange](#) attribute and their default value.

In order to provide some more sophisticated behavior, the [PhoneManagement](#) MAY support the following optional 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 [CMS], section 2.4.22, Relationships Between State Variables). If the TS supports this attribute, is CONDITIONALLY REQUIRED to support also the [GetSelectedValues\(\)](#) optional action.

## 2.4 State Variables

The [PhoneManagement](#) reuses some of the state variables defined in the [ConfigurationManagement](#) service.

The Table 2-2 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/Optional constraint for the *PhoneManagement*.

All the required state variables from the CMS MUST be implemented by *PhoneManagement*, as specified in [DEVICE]. It is up to the implementation to chose whether to support optional 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/Optionality constraint of the respective actions for the *PhoneManagement*. The A\_ARG\_TYPE\_ state variables are not listed in this document, refer to the [CMS] for the details.

**Table 2-2: State Variables for Eventing**

| Variable Name                      | CMS<br>R/O <sup>1</sup> | TS<br>R/O <sup>1</sup> |
|------------------------------------|-------------------------|------------------------|
| <i>ConfigurationUpdate</i>         | <u>R</u>                | <u>R</u>               |
| <i>CurrentConfigurationVersion</i> | <u>R</u>                | <u>R</u>               |
| <i>SupportedDataModelsUpdate</i>   | <u>R</u>                | <u>R</u>               |
| <i>SupportedParametersUpdate</i>   | <u>R</u>                | <u>R</u>               |
| <i>AttributeValuesUpdate</i>       | <u>O</u>                | <u>O</u>               |
| <i>InconsistentStatus</i>          | <u>O</u>                | <u>O</u>               |

<sup>1</sup> R = REQUIRED, O = OPTIONAL, CR = CONDITIONALLY REQUIRED, CO = CONDITIONALLY OPTIONAL, X = Non-standard, add -D when deprecated (e.g., R-D, O-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.*

## 2.5 Actions

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

The Table 2-3 below lists all the CMS actions, with the following additional information:

- The column “CMS R/O” indicates whether the action is required or optional in the [CMS]; all the required actions from the CMS MUST be implemented by the device supporting *PhoneManagement*.
- The column “TS R/O” indicates whether the action is either required or optional for *PhoneManagement*; some of the optional actions are required for the *PhoneManagement*;
- The column “TelCP R/O” indicates whether the TelCP MUST support the action to be fully compliant with the TS.

Refer to [CMS] for more detailed information on each actions, sub sections below give the additional information on some of the actions in *PhoneManagment* perspective.

**Table 2-3: Required Actions**

| Name  | CMS<br>R/O <sup>1</sup> | TS<br>R/O <sup>2</sup> | TelCP<br>R/O <sup>3</sup> |
|---|-------------------------|------------------------|---------------------------|
| <a href="#"><u>GetSupportedDataModels()</u></a>         | <u>R</u>                | <u>R</u>               | <u>R</u>                  |
| <a href="#"><u>GetSupportedParameters()</u></a>         | <u>R</u>                | <u>R</u>               | <u>R</u>                  |
| <a href="#"><u>GetInstances()</u></a>                   | <u>R</u>                | <u>R</u>               | <u>R</u>                  |
| <a href="#"><u>GetValues()</u></a>                      | <u>R</u>                | <u>R</u>               | <u>R</u>                  |
| <a href="#"><u>GetSelectedValues()</u></a>              | <u>O</u>                | <u>CR</u>              | <u>CR</u>                 |
| <a href="#"><u>SetValues()</u></a>                      | <u>O</u>                | <u>R</u>               | <u>R</u>                  |
| <a href="#"><u>CreateInstance()</u></a>                 | <u>O</u>                | <u>R</u>               | <u>R</u>                  |
| <a href="#"><u>DeleteInstance()</u></a>                 | <u>O</u>                | <u>R</u>               | <u>R</u>                  |
| <a href="#"><u>GetAttributes()</u></a>                  | <u>R</u>                | <u>R</u>               | <u>R</u>                  |
| <a href="#"><u>SetAttributes()</u></a>                  | <u>O</u>                | <u>O</u>               | <u>O</u>                  |
| <a href="#"><u>GetInconsistentStatus()</u></a>          | <u>O</u>                | <u>O</u>               | <u>O</u>                  |
| <a href="#"><u>GetConfigurationUpdate()</u></a>         | <u>R</u>                | <u>R</u>               | <u>R</u>                  |
| <a href="#"><u>GetCurrentConfigurationVersion()</u></a> | <u>R</u>                | <u>R</u>               | <u>R</u>                  |
| <a href="#"><u>GetSupportedDataModelsUpdate()</u></a>   | <u>R</u>                | <u>R</u>               | <u>R</u>                  |
| <a href="#"><u>GetSupportedParametersUpdate()</u></a>   | <u>R</u>                | <u>R</u>               | <u>R</u>                  |
| <a href="#"><u>GetAttributeValuesUpdate()</u></a>       | <u>O</u>                | <u>O</u>               | <u>O</u>                  |

<sup>1</sup> For a device this column indicates whether the action MUST be implemented or not, where R = REQUIRED, O = OPTIONAL, CR = CONDITIONALLY REQUIRED, CO = CONDITIONALLY OPTIONAL, X = Non-standard, add -D when deprecated (e.g., R-D, O-D)..

<sup>2</sup> For a device this column indicates whether the action MUST be implemented or not, where R = REQUIRED, O = OPTIONAL, CR = CONDITIONALLY REQUIRED, CO = CONDITIONALLY OPTIONAL, X = Non-standard, add -D when deprecated (e.g., R-D, O-D)..

<sup>3</sup> For a control point this column indicates whether a control point MUST be capable of invoking this action, where R = REQUIRED, O = OPTIONAL, CR = CONDITIONALLY REQUIRED, CO = CONDITIONALLY OPTIONAL, X = Non-standard, add -D when deprecated (e.g., R-D, O-D).

### 2.5.1 [GetSelectedValues\(\)](#)

If [PhoneManagment](#) implementation supports the [Version](#) attribute in the *Phone Data Model*, then the [GetSelectedValues\(\)](#) action MUST 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 section 3.3.3 in Theory of Operation (Informative) for detail mechanism for retrieving the changes.

### **2.5.2 SetValues()**

The SetValues() action is an optional action for the CMS, but the PhoneMangement perspective it is a required action and MUST 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 the section 3: Theory of Operation (Informative).

### **2.5.3 CreateInstance()**

The CreateInstance() action is an optional action for the CMS, but the PhoneMangement perspective it is a required action and MUST 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 the section 3: Theory of Operation (Informative).

### **2.5.4 DeleteInstance()**

The DeleteInstance() action is an optional action for the CMS, but the PhoneMangement perspective it is a required action and MUST 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 the section 3: Theory of Operation (Informative).

### **2.5.5 SetAttributes()**

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

The PhoneManagement defines some of the optional *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 MUST be supported in the implementation. Examples can be found in section 3: Theory of Operation (Informative).

### 3 Theory of Operation (Informative)

This section 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*).

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

#### 3.1 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*.

#### 3.2 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.

#### 3.3 Managing the *Address Book*

[\*The PhoneManagement\*](#) provides the set of functions for managing the *Address Book Parameters* defined in the table `/UPnP/PHONE/AddressBook/`.

##### 3.3.1 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 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 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>
```

### 3.3.2 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</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>
</cms:ParameterValueList>
```

### 3.3.3 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 identified 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.

### 3.3.4 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 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 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/Form  
attedName</ParameterPath>  
  <Value>Jane Doe Jr.</Value>  
</Parameter>  
<Parameter>  
  <ParameterPath>/UPnP/PHONE/AddressBook/Contact/25/Identification/Nick  
Name</ParameterPath>  
  <Value>Jane</Value>  
</Parameter>  
[. . .other parameters about the contact. . .]  
<Parameter>  
  <ParameterPath>/UPnP/PHONE/AddressBook/Contact/25/Explanatory/Sound/V  
alue</ParameterPath>  
  <Value>dzeLMakGA1UEBhMCVVMxLDA...qBgNVBAoTI05ldHNjYX</Value>
```

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

### 3.3.5 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>
```

### 3.3.6 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.

1. 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/
```

2. 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</Parameter
    rInitializationPath>
    <Value>Doe</Value>
  </Node>
  <Node>
    <ParameterInitializationPath>Identification/Name/GivenName</Parameter
    InitializationPath>
    <Value>John</Value>
  </Node>
</cms:ParameterInitialValueList>
```

### 3.3.7 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 must invoke the [DeleteInstance\(\)](#) action with the input argument [InstanceIdentifier](#) set to value as:

```
/UPnP/PHONE/AddressBook/21/
```

### 3.3.8 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>

```

### 3.3.9 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 must 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*.

## 3.4 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/
```

### 3.4.1 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 identified by the parameter `/UPnP/PHONE/Settings/RingingProfiles/Active`.

### 3.4.2 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</ParameterPath>
    <Value>Normal</Value>
  </Parameter>
  <Parameter>
    <ParameterPath>/UPnP/PHONE/Settings/RingingProfiles/Profile/3/Enable</ParameterPath>
    <Value>0</Value>

```

```

</Parameter>
[ . . . ]
<Parameter>
  <ParameterPath>/UPnP/PHONE/Settings/RingingProfiles/Profile/#/Message
s/Volume</ParameterPath>
  <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<
/ParameterPath>
  <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.

### 3.4.3 Creating a New Profile

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

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

This creates a new profile for the telephone, then the TelCP can use the [SetValues\(\)](#) action for setting the actual values of the *Parameters* for the profile.

### 3.4.4 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</ParameterPath>
    <Value>5</Value>
  </Parameter>
</cms:ParameterValueList>
```

### 3.4.5 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/
```

### 3.4.6 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.

## 3.5 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/
```

### 3.5.1 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 must use the [GetValues\(\)](#) or the [GetSelectedValues\(\)](#) action to retrieve the change in the `CurrentPowerSource` *Parameter*.

### 3.5.2 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
1 </ParameterPath>
  <Value>30</Value>
</Parameter>
</cms:ParameterValueList>
```

To receive some information when the value of CurrentPowerLevel goes below 30, the TelCP must also properly set its value of [EventOnChange](#) attribute to 1, using the [SetAttributes\(\)](#) action as explained in 3.5.1.

### 3.5.3 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.

## **4 XML Service Description**

The service description is specified in [CMS].

## Appendix A. Phone Data Model (Normative)

This section defines the data model for the [TelephonyServer:1](#) device called as the *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 [CMS].

The following rules are defined in accordance with [CMS] specification:

- All names in this table of *Parameter* definitions MUST be prefixed by /UPnP/PHONE/.
- Vendor specific *Parameters* to extend the capabilities of *Phone Data Model* MUST 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 [CMS]) MUST include a row with the following information:
  - **URI:** urn:UPnP:phone:1:pdm:1.
  - **Location:** /UPnP/PHONE/
  - **URL:** http://www.upnp.org/specs/phone/UPnP-phone-PhoneManagement-v1-Service.pdf
  - **Description:** Telephony: phone data model.
  - **SourceLocation:** Not applicable.

### Columns Description

As defined in [CMS], Appendix B, section 3.2, columns description is as follows:

- **Name:** white rows contain *Leaf* names, whereas yellow rows contain *StructurePath* fragments from the common prefix to the *SingleInstance* or *MultiInstance Node*.
- **Type:** the *Type* attribute value for *Leaf Nodes*, otherwise (yellow rows) it is specified whether the *Node* is *SingleInstance* or *MultiInstance*.
- **Acc.:** stands for *Access* attribute value of the *Node*. Possible values are “W” (the *Parameter* is writable, or the *Instance* is creatable) and “-“ (the *Parameter* is read only). If a *Parameter* is writable means that it makes sense to write (i.e. configure) it, and therefore does not mean that it must be writable for all implementations. The control point should use the [GetAttributes\(\)](#) action to verify what is implemented on the device. On the opposite side, if a *Parameter* is read only means that it does not make sense to write (i.e. configure) it, and therefore it must be read only for all implementations. Check with [CMS] for further details concerning this attribute.
- **Req.:** stands for Required. Possible values are “R” (the *Node* implementation is required), “O” (the *Node* implementation is optional) and “CR” (the *Node* implementation is conditionally required).
- **Description:** describes the *Parameter* meaning.
- **EOC:** stands for *EventOnChange*. Indicates whether the *EventOnChange* attribute is supported by the *Node* and its default value. The dash “-“ means that the support for that attribute is optional. 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). **Note:** Vendors can extend the list of the *Parameters* supporting the [EventOnChange](#) attribute.
- **Ver:** stands for *Version*. Indicates when the *Version* attribute is supported, whether the *Parameter* MUST also support (R) it. The dash “-“ means that the support for that attribute is optional. **Note:** Vendors can extend the list of the *Parameters* supporting the [Version](#) attribute.
-

## 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 MUST be used when `Value` contains a reference to the resource (URI). Other specific IANA registered formats MUST 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.

## Phone Data Model Overview

The following Figure 2 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.

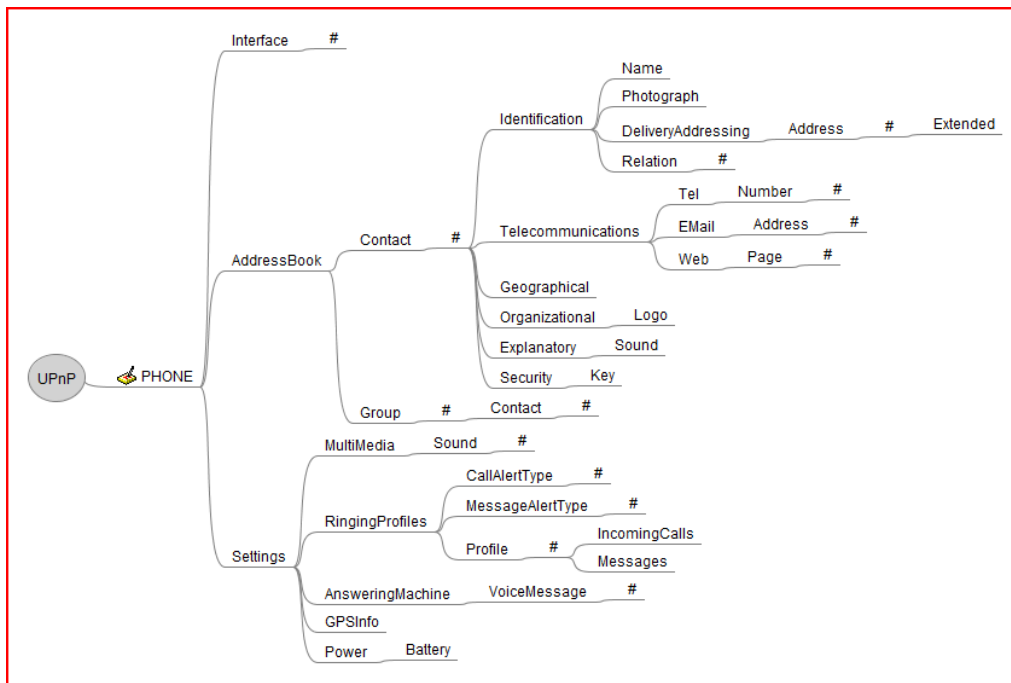


Figure 2: Overview of the *Phone Data Model*

## The Phone Data Model

| Name                     | Type                  | Acc. | Req | Description   | EOC | Ver |
|--------------------------|-----------------------|------|-----|---|-----|-----|
| /UPnP/PHONE/             | <i>SingleInstance</i> | -    | R   | The root node of the Phone Data Model.  | -   | -   |
| InterfaceNumberOfEntries | unsignedInt           | -    | CR  | Number of interfaces.<br>It MUST be provided when the /UPnP/PHONE/Interface/#/ is supported.  | 0   | R   |
| /UPnP/PHONE/Interface/#/ | <i>MultiInstance</i>  | -    | O   | 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).<br>Disabling and then enabling an interface causes the restart of all that interface and all the supported connections.                              | 0   | R   |
| Status                   | string                | -    | R   | Current operational status of this interface. Some of the allowed values are: <ul style="list-style-type: none"> <li>• "Initializing"</li> <li>• "Registering"</li> <li>• "Active"</li> <li>• "Unregistering"</li> <li>• "Error"</li> <li>• "Testing"</li> <li>• "Quiescent"</li> <li>• "Disabled"</li> </ul> | -   | -   |

| Name                     | Type                  | Acc. | Req | Description   | EOC | Ver |
|--------------------------|-----------------------|------|-----|---|-----|-----|
| Type                     | string                | -    | R   | The type of this interface. Some of the allowed values are: <ul style="list-style-type: none"> <li>• "GSM/GPRS"</li> <li>• "UMTS/WCDMA"</li> <li>• "HSPA"</li> <li>• "WiFi"</li> <li>• "Bluetooth"</li> <li>• "USB"</li> <li>• "Ethernet"</li> <li>• "PSTN/ISDN"</li> <li>• "VoIP"</li> </ul> | -   | -   |
| StartTime                | dateTime              | -    | R   | The absolute dateTime of the last successful restart of this interface.   | -   | -   |
| /UPnP/PHONE/AddressBook/ | <i>SingleInstance</i> | -    | R   | Container of Address Book management parameters   | -   | -   |
| MaxContactsSupported     | unsignedInt           | -    | R   | This parameter declares the maximum number of contacts (entries, rows) that the <i>Address Book</i> can contain. If there is no upper limit, the value declared must be 0.  | 0   | -   |
| MaxGroupsSupported       | unsignedInt           | -    | R   | This parameter declares the maximum number of groups (entries, rows) that the <i>Address Book</i> can contain. If there is no upper limit, the value declared must be 0.  | 0   | -   |
| StorageLocation          | string                | -    | R   | It defines whether the <i>Address Book</i> is locally stored on the TS or other options.<br>Possible values are the following:<br>"local"<br>"remote"   | 0   | -   |
| ContactNumberOfEntries   | unsignedInt           | -    | R   | Number of contacts in the AddressBook.  | 0   | R   |
| GroupNumberOfEntries     | unsignedInt           | -    | CR  | Number of groups.<br>It MUST 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   | <p>A table where each row is a contact. A contact can be created, updated, deleted, read. Each contact (row) must be defined in terms of its properties (i.e. parameters): vCard specification (RFC2426) is a source for such properties.</p> <p>Each contact in this table of contacts is uniquely identified by the <i>Instance</i> identifier (see the primary key description in ConfigurationManagement:1 Service).</p> <p>In case one or more of the following values is not defined, the empty string MUST be used to represent "NO value defined". Constraints are specified in specific parameters' description, when needed.</p> | -   | -   |
| /UPnP/PHONE/AddressBook/Contact/#/Identification/ | SingleInstance | -    | R   | This <i>SingleInstance</i> is used in the table of contacts to capture information associated with the identification and naming of this contact.  | -   | -   |
| FormattedName                                     | string(64)     | R    | R   | <p>Specifies the formatted text corresponding to the name of the contact.</p> <p>For example:<br/>Mr. John Q. Public\, Esq.</p> <p>Note that the COMMA character (ASCII decimal 44) is escaped as well as the "n" for the newline.</p> <p>[From [VCARD]]</p> <p>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.</p>   | 0   | -   |
| NickName  | string(64)     | W    | R   | <p>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.</p> <p>One or more text values separated by a COMMA character (ASCII decimal 44). For example: Jim,Jimmie.</p> <p>[From [VCARD]]</p>   | 0   | -   |
| Birthday  | dateTime       | W    | O   | Specifies the birth date of the contact.   | 0   | -   |
| Anniversary                                       | dateTime       | W    | O   | Anniversary for the contact.   | 0   | -   |
| FileAs  | string(64)     | W    | O   | 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 (RFC2426).   | 0   | -   |
| Note  | string(256)    | W    | O   | Free text for notes.   | 0   | -   |

| Name   | Type           | Acc. | Req | Description   | EOC | Ver |
|--|----------------|------|-----|---|-----|-----|
| RelationNumberOfEntries  | unsignedInt    | -    | CR  | Number instances in the Relation table.<br>It MUST be provided when the /UPnP/PHONE/AddressBook/Contact/#!/Identification/Relation/#!/ is supported.  | 0   | -   |
| /UPnP/PHONE/AddressBook/Contact/#!/Identification/Name/        | SingleInstance | -    | R   | This <i>SingleInstance</i> 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 [VCARD]]  | 0   | -   |
| GivenName  | string(64)     | W    | R   | The given name of the contact. [From [VCARD]]   | 0   | -   |
| MiddleName   | string(64)     | W    | O   | The additional (middle) name of the contact.  | 0   | -   |
| AdditionalName   | string(64)     | W    | O   | The additional name of the contact. [From [VCARD]]  | 0   | -   |
| HonorificPrefix  | string(64)     | W    | O   | The honorific prefix. For example: Mr, Dr, ... [From [VCARD]]   | 0   | -   |
| HonorificSuffix  | string(64)     | W    | O   | The honorific suffix. For example: Jr, Esq.,... [From [VCARD]]  | 0   | -   |
| /UPnP/PHONE/AddressBook/Contact/#!/Identification/Relation/#!/ | MultiInstance  | -    | O   | This <i>MultiInstance</i> contains a list of relations from the current contact to another contact in the <i>Address Book</i> , when there is any.  | -   | -   |
| Type   | string         | W    | R   | Values for this parameter are (the quotes must not be used when specifying the values for this parameter):<br>"Assistant",<br>"Spouse",<br>"Child",<br>"Parent",<br>"Familiar",<br>"Friend",<br>"Colleague",<br>"Other".<br><br>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   | -   |

| Name  | Type           | Acc. | Req | Description   | EOC | Ver |
|---|----------------|------|-----|---|-----|-----|
| ContactIdentifier   | unsignedInt    | W    | R   | The <i>Instance</i> identifier of the related contact:<br><br>If there is a contact in this <i>Address Book</i> which represents the desired relation, this ContactIdentifier MUST be the InstanceIdentifier of such contact.<br><br>If there is no contact in this <i>Address Book</i> which represents the desired relation, this ContactIdentifier MUST be 0 (zero).   | 0   | -   |
| /UPnP/PHONE/AddressBook/Contact#/Identification/Photograph/ | SingleInstance | -    | O   | 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 <a href="http://www.iana.org/assignments/media-types/">http://www.iana.org/assignments/media-types/</a> , are: <ul style="list-style-type: none"> <li>text/plain (<a href="http://www.iana.org/assignments/media-types/text/">http://www.iana.org/assignments/media-types/text/</a>).</li> <li>all image/... media types (<a href="http://www.iana.org/assignments/media-types/image/">http://www.iana.org/assignments/media-types/image/</a>).</li> </ul> [From [VCARD]] | 0   | -   |
| Value   | string(32k)    | W    | R   | The "value" of the photograph, depending on the MIMETYPE: <ul style="list-style-type: none"> <li>If the MIMETYPE is text/plain this parameter MUST contain the URI where the photograph is defined. For example:<br/>http://www.abc.com/pub/photos/jqpublic.gif</li> <li>If the MIMETYPE is image/... this parameter MUST contain the photograph encoded with respect to the image MIME type specified. For example, if the MIMETYPE is image/jpg the Value MUST contain the encoded jpg image.</li> </ul> [From [VCARD]]   | 0   | -   |
| /UPnP/PHONE/AddressBook/Contact#/DeliveryAddressing/        | SingleInstance | -    | O   | Specifies the components of the delivery address for the contact. This <i>SingleInstance</i> contains a table of address components (e.g. postal code, city,...).   | -   | -   |

| Name   | Type                | Acc. | Req | Description   | EOC | Ver |
|--|---------------------|------|-----|---|-----|-----|
| Preferred  | unsignedInt         | W    | R   | <p>The <i>Instance</i> identifier of the preferred address in the following list of Addresses. This MUST be an existing value.</p> <ul style="list-style-type: none"> <li>When there are NO Instances of Addresses, the value is 0 and can not be changed. An attempt to change that value MUST return the proper fault code (CMS::705 "Invalid Value").</li> <li>When there is only one <i>Instance</i> of Addresses, the value is the <i>Instance</i> identifier of such <i>Instance</i>. An attempt to set invalid values will result in the proper fault code returned by the device (CMS::705 "Invalid Value").</li> <li>When there are more than a single <i>Instance</i> 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 <i>Instance</i> identifier 3, 5 and 67, the control point can set this parameter to 3, 5 or 67. Any other values will produce an error.</li> </ul> <p>[From [VCARD]]</p> | 0   | -   |
| AddressNumberOfEntries   | unsignedInt         | -    | R   | Number of delivery addresses for this contact.  | 0   | -   |
| /UPnP/PHONE/AddressBook/Contact#/DeliveryAddressing/Address/#/ | <i>MultInstance</i> | R    | R   | <p>This <i>MultInstance</i> contains the addresses components of the contact.</p> <p>Where it makes semantic sense, individual text components of the following Address <i>MultInstance</i> children can include multiple text values (e.g., The "Street" parameter with multiple lines) separated by the COMMA character (ASCII decimal 44).</p>   | -   | -   |

| Name            | Type       | Acc. | Req | Description   | EOC | Ver |
|-----------------|------------|------|-----|---|-----|-----|
| Type            | string     | W    | R   | <p>Possible values for this parameter are (the quotes must not be used when specifying the values for this parameter):</p> <ul style="list-style-type: none"> <li>• "dom" to indicate a domestic delivery address;</li> <li>• "intl" to indicate an international delivery address;</li> <li>• "postal" to indicate a postal delivery address;</li> <li>• "parcel" to indicate a parcel delivery address;</li> <li>• "home" to indicate a delivery address for a residence;</li> <li>• "work" to indicate delivery address for a place of work;</li> <li>• "pref" to indicate the preferred delivery address when more than one address is specified.</li> </ul> <p>The value of this parameter MUST be at least (see below) one of the above types, and MUST 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").</p> <p>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.</p> <p>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".</p> <p>[From [VCARD]]</p> | 0   | -   |
| PostOfficeBox   | string(64) | W    | O   | The post office box. [From [VCARD]]   | 0   | -   |
| ExtendedAddress | string(64) | W    | O   | The extended address. [From [VCARD]]  | 0   | -   |
| StreetAddress   | string(64) | W    | R   | The street address. [From [VCARD]]  | 0   | -   |
| Locality        | string(64) | W    | R   | The name of the locality (e.g. village or city). [From [VCARD]]   | 0   | -   |
| Region          | string(64) | W    | R   | The name of the region (e.g. state or province) . [From [VCARD]]  | 0   | -   |
| PostalCode      | string(16) | W    | R   | The ZIP or postal code. [From [VCARD]]  | 0   | -   |
| Country         | string(64) | W    | R   | The country name (e.g. U.S.A.). [From [VCARD]]  | 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 <i>Instance</i> .<br><br>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:<br>"Mr.John Q. Public\n123 Main Street\nU.S.A"<br><br>Note that the COMMA character (ASCII decimal 44) is escaped as well as the "n" for the newline.<br>[From [VCARD]]  | 0   | -   |
| /UPnP/PHONE/AddressBook/Contact#/DeliveryAddressing/Address#/Extended/ | <i>SingleInstance</i> | -    | O   | 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/Contact#/Telecommunications/                   | <i>SingleInstance</i> | -    | R   | This <i>SingleInstance</i> concerns with information associated with the telecommunications addressing of the contact.   | -   | -   |
| /UPnP/PHONE/AddressBook/Contact#/Telecommunications/Tel/               | <i>SingleInstance</i> | -    | R   | This <i>SingleInstance</i> contains information associated with the contact telephone numbers for telephony communication.   | -   | -   |
| Preferred  | unsignedInt           | W    | R   | The <i>Instance</i> identifier of the preferred phone number in the following list of Numbers. This MUST be an existing value. <ul style="list-style-type: none"> <li>When there are NO Instances of Number, the value is 0 and can not be changed. An attempt to change that value MUST return the proper fault code (CMS::705 "Invalid Value").</li> <li>When there is only one <i>Instance</i> of Number, the value is the <i>Instance</i> identifier of such <i>Instance</i>. An attempt to set invalid values will result in the proper fault code returned by the device (CMS::705 "Invalid Value").</li> <li>When there are more than a single <i>Instance</i> 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 <i>Instance</i> identifier 3, 5 and 67, the control point can set this parameter to 3, 5 or 67. Any other values will produce an error.</li> </ul> [From [VCARD]] | 0   | R   |
| NumberOfEntries  | unsignedInt           | -    | R   | Number of telephone Numbers in the following table.  | -   | -   |

| Name  | Type                  | Acc. | Req | Description  | EOC | Ver |
|---|-----------------------|------|-----|--|-----|-----|
| /UPnP/PHONE/AddressBook/Contact#/Telecommunications/Tel/Number/#/ | <i>MultInstance</i>   | W    | R   | This <i>MultInstance</i> contains the Tel components of the contact.   | 0   | R   |
| Type  | string                | W    | R   | <p>Possible values for this parameter are (the quotes must not be used when specifying the values for this parameter):</p> <p>"Mobile", "Mobile(home)", "Mobile(business)",<br/> "Telephone", "Telephone(home)", "Telephone(business)",<br/> "Video", "Video(home)", "Video(business)",<br/> "Internet", "Internet(home)", "Internet(business)",<br/> "FAX", "FAX(home)", "FAX(business)",<br/> "Car-phone",<br/> "Pager",<br/> "BBS" to indicate a bulletin board system telephone number,<br/> "Modem" to indicate a MODEM connected telephone number,<br/> "ISDN" to indicate an ISDN service telephone number,<br/> "PCS" to indicate a personal communication services telephone number.</p> <p>The value of this parameter MUST be at least (see below) one of the above types, and MUST 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").</p> <p>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.</p> <p>[From [VCARD]]</p> | 0   | R   |
| Country   | string(4)             | W    | O   | The country international prefix (e.g. "+1" for the U.S.A.). This parameter is an Outlook extension to vCard specification (RFC2426).  | 0   | R   |
| Prefix  | string(4)             | W    | O   | The City/Area code (e.g. 206). This parameter is an Outlook extension to vCard specification (RFC2426).  | 0   | R   |
| Number  | string(64)            | W    | R   | The local telephone number. [From [VCARD]]   | 0   | R   |
| Extension   | string(64)            | W    | O   | The extension of the telephone number. This parameter is an Outlook extension to vCard specification (RFC2426).  | 0   | -   |
| /UPnP/PHONE/AddressBook/Contact#/Telecommunications/EMail/        | <i>SingleInstance</i> | -    | R   | Specifies the electronic mail addresses for communication with the contact.  | -   | -   |
| Mailer  | string(64)            | W    | O   | <p>The type of electronic mail software that is used by the individual associated with the contact. For example: "PigeonMail 2.1".</p> <p>[From [VCARD]]</p>   | 0   | -   |

| Name   | Type                | Acc. | Req | Description   | EOC | Ver |
|--|---------------------|------|-----|---|-----|-----|
| Preferred  | unsignedInt         | W    | R   | <p>The <i>Instance</i> identifier of the preferred mail address in the following list of addresses. This MUST be an existing value.</p> <ul style="list-style-type: none"> <li>When there are NO Instances of Address, the value is 0 and can not be changed. An attempt to change that value MUST return the proper fault code (CMS::705 "Invalid Value").</li> <li>When there is only one <i>Instance</i> of Address, the value is the <i>Instance</i> identifier of such <i>Instance</i>. An attempt to set invalid values will result in the proper fault code returned by the device (CMS::705 "Invalid Value").</li> <li>When there are more than a single <i>Instance</i> 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 <i>Instance</i> identifier 3, 5 and 67, the control point can set this parameter to 3, 5 or 67. Any other values will produce an error.</li> </ul> <p>[From [VCARD]]</p> | 0   | R   |
| WebAddress   | string(256)         | W    | O   | The web address. This parameter is an Outlook extension to vCard specification (RFC2426).   | 0   | -   |
| IMAddress  | string(256)         | W    | O   | Instant Messaging address. This parameter is an Outlook extension to vCard specification (RFC2426).   | 0   | R   |
| AddressNumberOfEntries   | unsignedInt         | -    | R   | Number of email addresses.  | 0   | -   |
| /UPnP/PHONE/AddressBook/Contact#/Telecommunications/EMail/Address/#/ | <i>MultInstance</i> | W    | R   | This <i>MultInstance</i> contains the e-mail addresses component of the contact.  | 0   | R   |
| Type   | string              | W    | R   | <p>Values for this parameter are (the quotes must not be used when specifying the values for this parameter):</p> <p>"Home", the address is for the whole home/family use</p> <p>"Business", the address is used for business</p> <p>"Personal", the address is for private uses</p> <p>"Other", any other purposes fro this address</p> <p>The value of this parameter MUST be at least (see below) one of the above types, and MUST 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").</p> <p>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.</p>  | 0   | R   |
| Address  | string(256)         | W    | R   | The internet e-mail address. [From [VCARD]]   | 0   | R   |



| Name   | Type           | Acc. | Req | Description  | EOC | Ver |
|--|----------------|------|-----|--|-----|-----|
| DisplayAs  | string(64)     | W    | O   | 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 (RFC2426).   | 0   | -   |
| /UPnP/PHONE/AddressBook/Contact#/Telecommunications/ Web/        | SingleInstance | -    | O   | Specifies the Web pages associated with the contact.   | -   | -   |
| Preferred  | unsignedInt    | W    | R   | <p>The <i>Instance</i> identifier of the preferred URL in the following list of Web Pages. This MUST be an existing value.</p> <ul style="list-style-type: none"> <li>When there are NO Instances of Pages, the value is 0 and can not be changed. An attempt to change that value MUST return the proper fault code (CMS::705 "Invalid Value").</li> <li>When there is only one <i>Instance</i> of Pages, the value is the <i>Instance</i> identifier of such <i>Instance</i>. An attempt to set invalid values will result in the proper fault code returned by the device (CMS::705 "Invalid Value").</li> <li>When there are more than a single <i>Instance</i> 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 <i>Instance</i> identifier 3, 5 and 67, the control point can set this parameter to 3, 5 or 67. Any other values will produce an error.</li> </ul> | 0   | -   |
| PageNumberOfEntries  | unsignedInt    | -    | R   | Number of web pages.   | 0   | -   |
| /UPnP/PHONE/AddressBook/Contact#/Telecommunications/ Web/Page/#/ | MultInstance   | W    | R   | This <i>MultInstance</i> contains the Web Pages component of the contact.  | -   | -   |
| Type   | string         | W    | R   | <p>Values for this parameter are (the quotes must not be used when specifying the values for this parameter):</p> <p>"Home", the address is for the whole home/family use</p> <p>"Business", the address is used for business</p> <p>"Personal", the address is for private uses</p> <p>"Other", any other purposes fro this address</p> <p>The value of this parameter MUST be at least (see below) one of the above types, and MUST 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").</p> <p>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.</p>  | 0   | -   |
| URL  | string(256)    | W    | O   | The Web address. It MUST be a valid URL.   | 0   | -   |

| Name  | Type           | Acc. | Req | Description   | EOC | Ver |
|---|----------------|------|-----|---|-----|-----|
| /UPnP/PHONE/AddressBook/Contact/#/Geographical/   | SingleInstance | -    | O   | This <i>SingleInstance</i> is concerned with information associated with geographical positions or regions associated with the contact  | -   | -   |
| TZ  | string(64)     | W    | O   | 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:<br>"05:00; EST; Raleigh/North America"<br>[From [VCARD]]   | 0   | -   |
| Geo   | string(64)     | W    | O   | 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 MUST 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:<br>decimal = degrees + minutes/60 + seconds/3600.<br>This is an example of Geo value:<br>"37.386013;-122.082932".<br>[From [VCARD]] | 0   | -   |
| /UPnP/PHONE/AddressBook/Contact/#/Organizational/ | SingleInstance | -    | O   | This <i>SingleInstance</i> 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:<br>"Director\, Research and Development"<br>Note that the COMMA character (ASCII decimal 44) is escaped with the back-slash \.<br>[From [VCARD]]  | 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:<br>"Programmer"<br>[From [VCARD]]   | 0   | -   |

| Name   | Type           | Acc. | Req | Description   | EOC | Ver |
|--|----------------|------|-----|---|-----|-----|
| Agent  | string(64)     | W    | R   | <p>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:</p> <ul style="list-style-type: none"> <li>• A single vCard value (RFC 2426). For example "BEGIN:VCARD\nFN: ...END:VCARD\n".</li> <li>• A reference to a contact in this AddressBook. For Example: "/UPnP/PHONE/AddressBook/Contact/45/". This extends the to vCard specification (RFC2426).</li> <li>• An URI to specify information outside of this scope. For example: "URI: CID:JQPUBLIC.part3.960129T083020.xyzMail@host3.com".</li> </ul> <p>[From [VCARD]]</p> | 0   | -   |
| Org  | string(64)     | W    | R   | <p>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.</p> <p>For example:"ABC\, Inc.;North American Division;Marketing".</p> <p>[From [VCARD]]</p>   | 0   | -   |
| /UPnP/PHONE/AddressBook/Contact/#/Organizational/Logo/ | SingleInstance | -    | O   | Graphic image of a logo associated with the contact.  | -   | -   |
| MIMETYPE   | string         | W    | R   | <p>The type of the following Value parameter. Allowed values, as a subset from the IANA registered formats <a href="http://www.iana.org/assignments/media-types/">http://www.iana.org/assignments/media-types/</a>, are:</p> <ul style="list-style-type: none"> <li>• text/plain (<a href="http://www.iana.org/assignments/media-types/text/">http://www.iana.org/assignments/media-types/text/</a>).</li> <li>• all image/... media types (<a href="http://www.iana.org/assignments/media-types/image/">http://www.iana.org/assignments/media-types/image/</a>).</li> </ul> <p>[From [VCARD]]</p>  | 0   | -   |
| Value  | string(32k)    | W    | R   | <p>The "value" of the logo, depending on the MIMETYPE:</p> <ul style="list-style-type: none"> <li>• If the MIMETYPE is text/plain this parameter MUST contain the URI where the contact's logo is defined. For example: <a href="http://www.abc.com/pub/logos/jqpublic.gif">http://www.abc.com/pub/logos/jqpublic.gif</a></li> <li>• If the MIMETYPE is image/... this parameter MUST contain the logo encoded with respect to the image MIME type specified. For example, if the MIMETYPE is image/jpg the Value MUST contain the encoded jpg logo image.</li> </ul> <p>[From [VCARD]]</p>   | 0   | -   |

| Name   | Type           | Acc. | Req | Description   | EOC | Ver |
|--|----------------|------|-----|---|-----|-----|
| /UPnP/PHONE/AddressBook/Contact/#/Explanatory/ | SingleInstance | -    | O   | This <i>SingleInstance</i> 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:<br>"TRAVEL AGENT" or<br>"INTERNET,IETF,INDUSTRY,INFORMATION TECHNOLOGY"<br>Are valid values for this parameter.<br>[From [VCARD]]          | 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:<br>"This fax number is operational 0800 to 1715 EST\, Mon-Fri".<br>[From [VCARD]]  | 0   | -   |
| Prod   | string(64)     | W    | O   | 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:<br>"-//ONLINE DIRECTORY//NONSGML Version 1//EN".<br>[From [VCARD]] | 0   | -   |
| Rev  | string(64)     | W    | O   | 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:<br>"1995-10-31T22:27:10Z".<br>[From [VCARD]]   | 0   | -   |

| Name       | <u>Type</u> | <u>Acc.</u> | <u>Req</u> | <u>Description</u>  | <u>EOC</u> | <u>Ver</u> |
|------------|-------------|-------------|------------|---|------------|------------|
| SortString | string(64)  | W           | R          | <p>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/.</p> <p>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.</p> <p>When this parameter value is assigned to a contact, then this family name or given name value is used for sorting among contacts.</p> <p>For example, in the case of family name sorting, the following examples define common sort string usage with the FormattedName and Name parameters.</p> <pre> /.../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 </pre> <p>[From [VCARD]]</p> | 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.<br>Example: "19950401-080045-40000F192713-0052".<br>[From [VCARD]]   | 0   | -   |
| URL  | string(256)    | W    | R   | Specifies a uniform resource locator associated with this contact. [From [VCARD]]  | 0   | -   |
| /UPnP/PHONE/AddressBook/Contact/#/Explanatory/Sound/ | SingleInstance | -    | O   | This <i>SingleInstance</i> 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 <a href="http://www.iana.org/assignments/media-types/">http://www.iana.org/assignments/media-types/</a> , are: <ul style="list-style-type: none"> <li>text/plain (<a href="http://www.iana.org/assignments/media-types/text/">http://www.iana.org/assignments/media-types/text/</a>).</li> <li>all audio/... media types (<a href="http://www.iana.org/assignments/media-types/audio/">http://www.iana.org/assignments/media-types/audio/</a>).</li> </ul> [From [VCARD]]                      | 0   | -   |
| Value  | string(32k)    | W    | R   | The "value" of the sound, depending on the MIMETYPE: <ul style="list-style-type: none"> <li>If the MIMETYPE is text/plain this parameter MUST contain the URI where the contact's sound is defined. For example: <a href="http://www.abc.com/pub/thiscontact/pronunciation.mpeg">http://www.abc.com/pub/thiscontact/pronunciation.mpeg</a></li> <li>If the MIMETYPE is audio/... this parameter MUST contain the sound encoded with respect to the audio MIME type specified. For example, if the MIMETYPE is audio/mp4 the Value MUST contain the encoded mp4 audio stream.</li> </ul> [From [VCARD]] | 0   | -   |
| /UPnP/PHONE/AddressBook/Group/#/                     | MultiInstance  | W    | O   | 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   | <p>Values for this parameter are (the quotes must not be used when specifying the values for this parameter):</p> <p>“Friends”,<br/> “Familiars”,<br/> “Parents”,<br/> “Children”,<br/> “Colleagues”,<br/> “Organization”,<br/> “Group”,<br/> “Sport”,<br/> “Hobby”,<br/> “Other”.</p> <p>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.</p>  | 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/#/ | <i>MultiInstance</i>  |      | 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   | <p>The <i>Instance</i> identifier of the contact belonging to the group. This MUST be an existing valid contact.</p> <p>If the related contact is deleted, this <i>Instance</i> MUST be deleted.</p>  | 0   | R   |
| /UPnP/PHONE/Settings/MultiMedia/           | <i>SingleInstance</i> |      | O   | Multi Media folder containing images and sounds.  | -   | -   |
| SoundNumberOfEntries                       | unsignedInt           | -    | R   | Number of multimedia sounds.  | -   | -   |
| /UPnP/PHONE/Settings/MultiMedia/Sound/#/   | <i>MultiInstance</i>  | W    | R   | <p>Folder for sounds and melodies.</p> <p>Device implementations, accordingly with the Access attribute definition in [CMS], MAY provide some restrictions as follows:</p> <ul style="list-style-type: none"> <li>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.</li> <li>The device might deny the attempt to create a new Instance. In this case, the Access attribute value for this Instance MUST be ReadOnly instead of ReadWrite. See Access attribute in [CMS].</li> </ul> | -   | -   |

| Name                                  | Type                  | Acc. | Req | Description   | EOC | Ver |
|---------------------------------------|-----------------------|------|-----|---|-----|-----|
| 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 <a href="http://www.iana.org/assignments/media-types/">http://www.iana.org/assignments/media-types/</a> , are: <ul style="list-style-type: none"> <li>text/plain (<a href="http://www.iana.org/assignments/media-types/text/">http://www.iana.org/assignments/media-types/text/</a>).</li> <li>all audio/... media types (<a href="http://www.iana.org/assignments/media-types/audio/">http://www.iana.org/assignments/media-types/audio/</a>).</li> </ul>              | -   | -   |
| Value                                 | string(32k)           | W    | R   | The "value" of the sound, depending on the MIMEType: <ul style="list-style-type: none"> <li>If the MIMEType is text/plain this parameter MUST contain the URI where the melody is defined. For example: <a href="http://www.abc.com/pub/thiscontact/pronunciation.mpeg">http://www.abc.com/pub/thiscontact/pronunciation.mpeg</a></li> <li>If the MIMEType is audio/... this parameter MUST contain the melody encoded with respect to the audio MIME type specified. For example, if the MIMEType is audio/mp4 the Value MUST contain the encoded mp4 audio stream.</li> </ul> | -   | -   |
| UserDefined                           | boolean               | R    | O   | This optional parameter, when supported, can be used to distinguish among the user defined sounds and the factory default sounds.   | -   | -   |
| /UPnP/PHONE/Settings/RingingProfiles/ | <i>SingleInstance</i> | -    | O   | 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.   | -   | -   |
| Active                                | unsignedInt           | -    | R   | The <i>Instance</i> 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 MUST exist in each implementation. In case there is just one profile it MUST be active.<br>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    | O   | This parameter is used to test the ringing tone of the currently active profile.<br>When the Test parameter is set to true (1) the device MUST 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.<br>When the Test parameter is set to false (0) the device MUST 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.   | -   | -   |



| Name   | Type                | Acc. | Req | Description  | EOC | Ver |
|--|---------------------|------|-----|--|-----|-----|
| MessageAlertTypeNumberOfEntries                          | unsignedInt         | -    | R   | Number of message alert types.   | -   | -   |
| ProfileNumberOfEntries                                   | unsignedInt         | -    | R   | Number of profiles.  | -   | -   |
| /UPnP/PHONE/Settings/RingingProfiles/CallAlertType/#/    | <i>MultInstance</i> | -    | 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/#/ |                     |      |     | 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.  | -   | -   |
| /UPnP/PHONE/Settings/RingingProfiles/Profile/#/          | <i>MultInstance</i> | W    | R   | <p>Table representing the PHONE profiles.</p> <p>Device implementations, accordingly with the Access attribute definition in CMS, MAY provide some restrictions to profiles as follows:</p> <ul style="list-style-type: none"> <li>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.</li> <li>The device might deny the attempt to create a new Instance. In this case, the Access attribute value for this Instance MUST be ReadOnly instead of ReadWrite. See Access attribute in CMS.</li> </ul> <p>There MUST be at least one (default) profile in this table.</p> <p>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.</p> | -   | -   |

| Name  | Type           | Acc. | Req | Description  | EOC | Ver |
|---|----------------|------|-----|--|-----|-----|
| Name  | string         | W    | R   | <p>The name of the profile. Possible values for this profile name might be (the quotes must not be used when specifying the values for this parameter):</p> <p>“Normal”,<br/> “Silent”,<br/> “Party”,<br/> “Offline”.</p> <p>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”.</p>  | -   | -   |
| Active  | boolean        | W    | R   | <p>This indicates which is the active profile. The following rules MUST be obey by all device implementations:</p> <ul style="list-style-type: none"> <li>• There MUST be exactly one active profile at the same time. Its Active value is 1.</li> <li>• As the selected profile becomes active, all the remaining ones MUST be set to 0 (not active).</li> <li>• As an active profile is deleted (when permitted), the implementation will choose the new active profile among the other ones.</li> </ul> <p>As the Active profile changes, the telephone MUST update the value of /UPnP/PHONE/ Settings/RingingProfiles/Active parameter.</p> <p>As the control point changes the active profile, the device might announce the event to the user by playing the incoming calls’ alert tone.</p> | -   | -   |
| /UPnP/PHONE/Settings/RingingProfiles/Profile/#/IncomingCalls/ | SingleInstance | -    | R   | Contains the basic setting for the incoming calls of this profile.   | -   | -   |
| CallAlertType   | unsignedInt    | W    | R   | <p>Instance identifier representing the type of the alert. The types supported by the device MUST be declared in the table /UPnP/PHONE/ Settings/RingingProfiles/CallAlertTypes/#/. The control point MUST 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.</p>   | 0   | R   |
| RingTone  | unsignedInt    | W    | R   | <p>The Instance identifier of the /UPnP/PHONE/ Settings/MultiMedia/Sounds/#/, where the melody is took from. The control point MUST 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.</p> <p>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.</p>   | -   | -   |

| Name   | Type           | Acc. | Req | Description   | EOC | Ver |
|--|----------------|------|-----|---|-----|-----|
| Volume   | unsignedInt    | W    | R   | A number specifying the percentage of the maximum volume for the device. Allowed values are: <ul style="list-style-type: none"> <li>From 1 to 10, for volume levels from 10% to 100% of the maximum.</li> <li>0, from the mute option.</li> </ul>   | 0   | R   |
| /UPnP/PHONE/Settings/RingingProfiles/Profile/~/Messages/ | SingleInstance | -    | R   | Contains the basic setting for the incoming calls of this profile.  | -   | -   |
| MessageAlertType   | unsignedInt    | W    | R   | <i>Instance</i> identifier representing the type of the alert. The types supported by the device MUST be declared in the table /UPnP/PHONE/Settings/RingingProfiles/MessageAlertTypes/~/ . The control point MUST 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 <i>Instance</i> identifier of the /UPnP/PHONE/ Settings/MultiMedia/Sounds/~/ , where the melody is taken from. The control point MUST 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.<br><br>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 MUST 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: <ul style="list-style-type: none"> <li>From 1 to 10, for volume levels from 10% to 100% of the maximum.</li> <li>0, from the mute option.</li> </ul>   | 0   | R   |
| /UPnP/PHONE/Settings/AnsweringMachine/                   | SingleInstance | -    | O   | 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 <i>MultiInstance</i> node /UPnP/PHONE/AnsweringMachine/VoiceMessage/~/ . The control point can select the desired voice message and choose whether it must 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   |

| Name   | Type                  | Acc. | Req | Description  | EOC | Ver |
|--|-----------------------|------|-----|--|-----|-----|
| SelectedVoiceMessage                                   | unsignedInt           | W    | R   | <i>Instance</i> identifier representing the selected VoiceMessage for the answering machine. The voice messages MUST exist in the table /UPnP/PHONE/AnsweringMachine/VoiceMessage/#!/. The control point MUST 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   |
| /UPnP/PHONE/Settings/AnsweringMachine/VoiceMessage/#!/ | <i>MultInstance</i>   | W    | R   | Folder for recorded voice messages.<br>Device implementations, accordingly with the Access attribute definition in [CMS], MAY provide some restrictions as follows: <ul style="list-style-type: none"> <li>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.</li> <li>The device might deny the attempt to create a new Instance. In this case, the Access attribute value for this Instance MUST be ReadOnly instead of ReadWrite. See Access attribute in [CMS].</li> </ul> | -   | -   |
| 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 <a href="http://www.iana.org/assignments/media-types/">http://www.iana.org/assignments/media-types/</a> , are: <ul style="list-style-type: none"> <li>text/plain (<a href="http://www.iana.org/assignments/media-types/text/">http://www.iana.org/assignments/media-types/text/</a>).</li> <li>all audio/... media types (<a href="http://www.iana.org/assignments/media-types/audio/">http://www.iana.org/assignments/media-types/audio/</a>).</li> </ul>   | -   | -   |
| Value  | string(32k)           | W    | R   | The "value" of the voice message, depending on the MIMETYPE: <ul style="list-style-type: none"> <li>If the MIMETYPE is text/plain this parameter MUST contain the URI where the voice message is defined. For example: <a href="http://www.abc.com/pub/thiscontact/message.mpeg">http://www.abc.com/pub/thiscontact/message.mpeg</a></li> <li>If the MIMETYPE is audio/... this parameter MUST contain the message encoded with respect to the audio MIME type specified. For example, if the MIMETYPE is audio/mp4 the Value MUST contain the encoded mp4 audio stream.</li> </ul>  | -   | -   |
| /UPnP/PHONE/Settings/GPSInfo/                          | <i>SingleInstance</i> | -    | O   | 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 <a href="http://en.wikipedia.org/wiki/Latitude">http://en.wikipedia.org/wiki/Latitude</a> . The latitude is represented in decimal degrees (see: <a href="http://en.wikipedia.org/wiki/Decimal_degrees">http://en.wikipedia.org/wiki/Decimal_degrees</a> ) having 5 fixed decimal places accuracy (1.11 m).<br><br>To encode the value in the SOAP "int" standard type, it must be multiplied by 100000. For example, the decimal degree representation of the longitude 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 <a href="http://en.wikipedia.org/wiki/Longitude">http://en.wikipedia.org/wiki/Longitude</a> . The longitude is represented in decimal degrees (see: <a href="http://en.wikipedia.org/wiki/Decimal_degrees">http://en.wikipedia.org/wiki/Decimal_degrees</a> ) having 5 fixed decimal places accuracy (1.11 m).<br><br>To encode the value in the SOAP "int" standard type, it must be multiplied by 100000. For example, the decimal degree representation of the longitude 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/         | <i>SingleInstance</i> | -    | O   | Provides information regarding power source.  | -   | -   |
| CurrentPowerSource                  | string                | -    |     | The current power source type. Possible values are:<br>"AC Power"<br>"Battery"  | 1   | R   |
| /UPnP/PHONE/Settings/Power/Battery/ | <i>SingleInstance</i> | -    | O   | Provides information regarding the battery.   | -   | -   |
| Status                              | String                | -    |     | The current status of the battery. Possible values are:<br>"Available" means battery installed and working<br>"Charging" means battery installed and under charging<br>"Unavailable" means battery not installed<br>"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 the 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   |