ETSI TS 103 544-10 V1.3.1 (2019-10)

Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink®; Part 10: UPnP Client Profile Service

CAUTION

The present document has been submitted to ETSI as a PAS produced by CCC and approved by the ETSI Technical Committee Intelligent Transport Systems (ITS).

CCC is owner of the copyright of the document CCC-TS-026 and/or had all relevant rights and had assigned said rights to ETSI on an "as is basis". Consequently, to the fullest extent permitted by law, ETSI disclaims all warranties whether express, implied, statutory or otherwise including but not limited to merchantability, non-infringement of any intellectual property rights of third parties. No warranty is given about the accuracy and the completeness of the content of the present document.

Reference
RTS/ITS-98-10

Keywords
interface, ITS, PAS, smartphone

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

The present document can be downloaded from:
http://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at www.etsi.org/deliver.

Users of the present document should be aware that the document may be subject to revision or change of status.
Information on the current status of this and other ETSI documents is available at
https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

Copyright Notification

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.
The content of the PDF version shall not be modified without the written authorization of ETSI.
The copyright and the foregoing restriction extend to reproduction in all media.
©ETSI 2019.
© Car Connectivity Consortium 2011-2019.
All rights reserved.
ETSI logo is a Trade Mark of ETSI registered for the benefit of its Members.
MirrorLink® is a registered trademark of Car Connectivity Consortium LLC.
RFB® and VNC® are registered trademarks of RealVNC Ltd.
UPnP® is a registered trademark of Open Connectivity Foundation, Inc.
Other names or abbreviations used in the present document may be trademarks of their respective owners.
DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members.
3GPP™ and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.
oneM2M™ logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners.
GSM® and the GSM logo are trademarks registered and owned by the GSM Association.

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Intelligent Transport Systems (ITS).

The present document is part 10 of a multi-part deliverable. Full details of the entire series can be found in part 1 [i.1].

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The present document is part of the MirrorLink® specification which specifies an interface for enabling remote user interaction of a mobile device via another device. The present document is written having a vehicle head-unit to interact with the mobile device in mind, but it will similarly apply for other devices, which provide a color display, audio input/output and user input mechanisms.

The *TmClientProfile* service is a UPnP service that allows control points to register client profiles with the MirrorLink Server device and notifies it regarding MirrorLink Client preferences, settings and capabilities, which is be used for governing the interaction between the MirrorLink Server and the Client.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

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

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

The following referenced documents are necessary for the application of the present document.

[1] UPnP™ Forum: "UPnP™ Device Architecture 1.1", 15 October 2008.

NOTE: Available at http://upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.1.pdf.

[2] ETSI TS 103 544-2 (V1.3.1): "Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink®; Part 2: Virtual Network Computing (VNC) based Display and Control".

[3] ETSI TS 103 544-3 (V1.3.1): "Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink®; Part 3: Audio".

[4] ETSI TS 103 544-26 (V1.3.1): "Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink®; Part 26: Consumer Experience Principles and Basic Features".

[5] W3C Recommendation 11 April 2013: "XML Signature Syntax and Processing Version 1.1".

NOTE: Available at http://www.w3.org/TR/xmldsig-core/.

[6] Unicode Consortium: "Unicode 12.1 Character Code Charts".

NOTE: Available at http://www.unicode.org/charts/.

[7] ETSI TS 103 544-4 (V1.3.1): "Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink®; Part 4: Device Attestation Protocol (DAP)".

[8] IETF RFC 5280: "Internet X.509 Public Key Infrastructure Certificate", May 2008.

NOTE: Available at http://tools.ietf.org/html/rfc5280.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

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

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI TS 103 544-1 (V1.3.1): "Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink®; Part 1: Connectivity".

3 Definition of terms, symbols and abbreviations

3.1 Terms

Void.

3.2 Symbols

Void.

3.3 Abbreviations

Void.

4 Service Modeling Definitions

4.1 Service Type

The following service type identifies a service that is compliant with the present document:

urn:schemas-upnp-org:service:*TmClientProfile:1*.

TmClientProfile service is used herein to refer to this service type. The *TmClientProfile* service shall follow defined UPnP behaviour within the UPnP Device Architecture 1.1 [1].

4.2 State Variables

4.2.1 State Variable Overview

Table 4-1: Service State Variables

Variable Name	Req. or Opt.	Data Type	Allowed Value	Default Value	Eng. Units
UnusedProfileIDs	R	string	Undefined	Empty string	N/A
A_ARG_TYPE_ClientProfile	R	string	Undefined	Empty string	N/A
A_ARG_TYPE_ProfileID	R	ui4	Undefined	0	N/A
A_ARG_TYPE_String	R	string	Undefined	Empty string	N/A

Variable Name	Req. or Opt.	Data Type	Allowed Value	Default Value	Eng. Units
A_ARG_TYPE_INT	R	integer	Undefined	0	N/A
A_ARG_TYPE_Bool	R	string	true | false	false	N/A
MaxNumProfiles	R	ui2	Undefined	1	N/A
R = REQUIRED. O = OPTIONAL. X = Non-standard.					

4.2.2 UnusedProfileIDs

A string formatted as UTF-8 which consists of a comma separated list of *profileIDs* that are currently not being used by any MirrorLink service hosted on the device. Each entry in the list is of type *A_ARG_TYPE_ProfileID*.

This state variable is evented hence, any MirrorLink UPnP Control Point is notified using eventing mechanisms whenever the list of unused *profileIDs* changes.

Support for multiple Client Profiles reserved for future use, therefore only *profileID*=0 shall be initially available.

4.2.3 A_ARG_TYPE_ClientProfile

A string formatted as UTF-8 XML represents identification and monitoring capability information of the MirrorLink Client. Its structure is given in Table 4-2.

Table 4-2: Structure of A_ARG_TYPE_ClientProfile

Element	Description	Parent	Availability
clientProfile	Profile Information about a MirrorLink Client	-	Mandatory
clientID	ID of the MirrorLink Client (A_ARG_TYPE_String)	clientProfile	Mandatory
friendlyName	Short user-friendly description of client (A_ARG_TYPE_String)	clientProfile	Optional
manufacturer	Manufacturer Name (A_ARG_TYPE_String)	clientProfile	Mandatory
modelName	Model Name (A_ARG_TYPE_String)	clientProfile	Optional
modelNumber	Model Number (A_ARG_TYPE_String)	clientProfile	Mandatory
iconPreference	Desired properties for icons delivered from the MirrorLink Server device	clientProfile	Optional
mimetype	Type of icon image (A_ARG_TYPE_String) **Default:** "image/png"	iconPreference	Optional
width	Width of icon (A_ARG_TYPE_INT) **Default:** "128"	iconPreference	Optional
height	Height of icon (A_ARG_TYPE_INT) **Default:** "128"	iconPreference	Optional
depth	Color depth of icon (A_ARG_TYPE_INT) **Default:** "24"	iconPreference	Optional
connectivity	Client Connectivity settings	clientProfile	Optional

Element	Description	Parent	Availability
bluetooth+	Bluetooth settings	connectivity	Optional
bdAddr	Bluetooth MAC address (BD_ADDR). Indicates device support for Bluetooth on the MirrorLink Client. (A UTF-8 encoded string representing an unsigned 48-bit integer in hexadecimal format (without any "0x" prefix).)	bluetooth	Optional
startConnection	A_ARG_TYPE_Bool Bluetooth Connection will be initiated from MirrorLink Client **Default:** "true"	bluetooth	Optional
wifi	WiFi settings of the device	connectivity	Optional
macAddr	WiFi MAC address (A UTF-8 encoded string representing an unsigned 48-bit integer in hexadecimal format (without any "0x" prefix, and without any grouping using ":", "." or "-")	wifi	Mandatory
ssid	Service Set Identifier (SSID), Base64 encoded (A_ARG_TYPE_String)	wifi	Optional
roles	Comma separated list of supported roles. Allowed values are • AP (Access Point role) • Client (Client role) • P2P (Infrastructure-less) (A_ARG_TYPE_String) **Default:** AP,Client,P2P	wifi	Optional
protectionList	List of WiFi access protection	wifi	Optional
protection*	Access protection	protectionList	Optional
protocol	Security protocol used to protect WiFi access. Allowed values are • WEP • WPA • WPA2 • WPS NOTE: WEP/WPA is listed for legacy reasons, and should not be used (A_ARG_TYPE_String)	protection	Mandatory
passkey	Passkey/Shared key, Base64 encoded Shall be left empty, if transmitted over an unprotected or shared transport channel (e.g. WiFi) (A_ARG_TYPE_String)	protection	Mandatory
rtpStreaming	RTP streaming parameters (optional in ≤ MirrorLink 1.2)	clientProfile	Mandatory
payloadType	Comma separated list of supported RTP playload types. (A_ARG_TYPE_String) **Default:** "99"	rtpStreaming	Optional
audioIPL	Audio Initial Playback Latency in reference to payload type 99. (A_ARG_TYPE_INT) **Default:** "4800"	rtpStreaming	Optional

Element	Description	Parent	Availability
audioMPL	Audio Maximum Playback Length in reference to payload type 99. (A_ARG_TYPE_INT) **Default:** "9600"	rtpStreaming	Optional
lssMax	Maximum Latency Switched Sources time as defined in [3]. Value is in ms. (A_ARG_TYPE_INT) (≥ MirrorLink 1.3)	rtpStreaming	Mandatory
lssAvg	Average Latency Switched Sources time as defined in [3]. Value is in ms. (A_ARG_TYPE_INT) (≥ MirrorLink 1.3)	rtpStreaming	Mandatory
contentRules	Application UI content rules **Depreciated**	clientProfile	Deprecated
rule*	User Interface property **Depreciated**	contentRules	Deprecated
ruleId	Rule identifier **Depreciated** (A_ARG_TYPE_INT)	rule	Deprecated
ruleValue	Specific value **Depreciated** (A_ARG_TYPE_String)	rule	Deprecated
services	Profile information on different services supported from the client	clientProfile	Optional
notification	Configuration of the UPnP TmNotifi-cationService	services	Optional
notiUiSupport	Support for native notification UI (A_ARG_TYPE_Bool) **Default:** "false"	notification	Optional
maxActions	Maximum number of actions Shall be equal or greater than 2. MirrorLink Client should support at least 2 actions. (A_ARG_TYPE_INT) **Default:** "2"	notification	Optional
actionName MaxLength	Maximum supported length of the action name Shall be equal or greater than 10. (A_ARG_TYPE_INT) **Default:** "10"	notification	Optional
notiTitle MaxLength	Maximum supported length of the notification title. Shall be equal or greater than 20. (A_ARG_TYPE_INT) **Default:** "20"	notification	Optional
notiBody MaxLength	Maximum supported length of the notification body. Shall be equal or greater than 80. (A_ARG_TYPE_INT) **Default:** "80"	notification	Optional
mirrorLink Version	MirrorLink Client version	clientProfile	Mandatory

Element	Description	Parent	Availability
majorVersion	Major Version A_ARG_TYPE_INT	mirrorLink Version	Mandatory
minorVersion	MinorVersion A_ARG_TYPE_INT	mirrorLink Version	Mandatory
presentations	Presentation protocols supported from the MirrorLink Client. (≥ MirrorLink 1.2; optional in MirrorLink 1.2)	clientProfile	Mandatory
presentation	Comma-separated list of presentation protocols supported from the MirrorLink Client. • `hsml` • `wfd` • `vncu` • `vncw` (A_ARG_TYPE_String) **Default:** "`vncu`"	presentations	Mandatory
misc	Miscellaneous information (≥ MirrorLink 1.2; optional in MirrorLink 1.2)	clientProfile	Mandatory
driverDistractionSupport	Boolean flag, which Indicates whether MirrorLink Client supports driver distraction regulation. (A_ARG_TYPE_Bool) **Default:** "`true`"	misc	Mandatory
mlUiMode	Supported MirrorLink modes from the MirrorLink Client. (≥ MirrorLink 1.3)	misc	Mandatory
mode*	Supported MirrorLink mode. Allowed values are: • `immersive` • `classic` (A_ARG_TYPE_String)	mlUiMode	Mandatory
mlUiControl	Supported UI control mechanisms from the MirrorLink Client. (≥ MirrorLink 1.3)	misc	Mandatory
control*	Supported UI control mechanisms. Allowed values are: • `pointer` • `touch` • `knob` • `vc_bt_hfp` • `vc_rtp` • `ptt` (A_ARG_TYPE_String)	mlUiControl	Mandatory
serverInfo	Server status information presented from the MirrorLink Client. (≥ MirrorLink 1.3)	misc	Mandatory
info*	Server status information shown from MirrorLink Client. Allowed values are: • `battery` • `time` • `operator` • `signalStrength` • `none` (A_ARG_TYPE_String)	serverInfo	Mandatory

Element	Description	Parent	Availability
localization	Provide information about the localization support from the MirrorLink Client. (≥ MirrorLink 1.2)	clientProfile	Optional
characterSet	Comma-separated list of entry points into the UniCode Character Code Charts, which are supported from the MirrorLink Client device. (UTF-8 encoded string; each entry point is given in hexadecimal format (with "0x" prefix).	localization	Mandatory
certificates	Certification certificates. (≥ MirrorLink 1.3)	clientProfile	N/A for ML Server Mandatory for ML Client
clientDevice	X.509v3 certificate [8] issued during MirrorLink device certification. Certificate shall be validated using public key from *clientCA*. The certificate is Base64 encoded from ASN.1 DER format. (A_ARG_TYPE_String)	certificates	N/A for ML Server Mandatory for ML Client
clientCA	X.509v3 certificate [8] issued by the Ceritifcate Authority. Certificate shall be validated using public Certificate Authority's root key. The certificate is Base64 encoded from ASN.1 DER format. (A_ARG_TYPE_String)	certificates	N/A for ML Server Mandatory for ML Client
Signature	XML signature over entire contents of the clientProfile element. This is done as specified in [5]. The key used in calculating the signature shall be the private part of the application-specific key which public part was bound to the attestation of UPnP-Server component. (The public part can be used to verify the signature.) The *Reference* element of the XML signature shall point to device element. The *SignatureMethod* shall be RSA with SHA1. The *KeyInfo* element may be omitted. The mechanism for generation, exchange and maintenance of keys is out of scope for the present document. (≥ MirrorLink 1.2)	clientProfile	Mandatory for MirrorLink Server N/A for MirrorLink Client

The elements marked with a (*) can have multiple instances.

Implementation Note

The *manufacturer* of some older MirrorLink Client devices may be unavailable.

The *clientID* is a unique identifier of the MirrorLink Client device instance; this identifier shall be unique for all device instances from the device manufacturer given in the *manufacturer* element. It shall survive the MirrorLink Client's shut down and reboot. If may be reset during a factory reset.

Implementation Note

The *clientID* of some older MirrorLink Client devices need not be unique for all device instances.

The *modelNumber* is a unique number identifying a family of devices, which expose identical MirrorLink related behavior, from the device manufacturer given in the manufacturer element. The model number format is vendor specific. It shall be smaller than 32 bytes. The modelNumber values are recorded by the CCC Certification Body.

Implementation Note

Some older MirrorLink Client devices need not provide a (unique) model number.

UPnP Client Profile Service shall provide a valid MirrorLink minor and major version, not higher than the supported version in the UPnP TmServerDevice device description. The MirrorLink Server device shall use the MirrorLink version provided from the UPnP Client Profile Service.

Implementation Note

MirrorLink 1.1 Clients may not provide a *mirrorLinkVersion*.entry, as this entry has been optional in MirrorLink 1.1.

A MirrorLink Server and Client shall not include any elements within the *A_ARG_TYPE_ClientProfile*, which are not defined for the respective MirrorLink version of the connected device.

For deprecated values, the MirrorLink Client shall not include them into the UPnP Client Profile. The MirrorLink Server shall ignore any content provided in deprecated elements.

The MirrorLink Client shall provide a Bluetooth MAC address (*bdAddr*) using the UPnP Client Profile Service, if the MirrorLink Client has a Bluetooth module and cannot initiate a Bluetooth connection (*startConnection* = "false"). The MirrorLink Client shall always provide a Bluetooth MAC address (bdAddr), if the MirrorLink Client supports the UPnP Client Profile Service and the MirrorLink Client has a Bluetooth module, even if that module is not used within a potential MirrorLink connection.

The following applies from MirrorLink 1.2 onwards.

The MirrorLink Client may provide the MirrorLink Server an indication, within the *driverDistractionSupport* element, whether the client's display is potentially subject to driver distraction regulation or not. E.g. if the MirrorLink Client display is driver facing, the MirrorLink Client shall enable driver distraction regulation (*driverDistractionSupport* = "true"). In case a MirrorLink Client is placed in the rear seats for passengers, the MirrorLink Client is not required to apply driver distraction regulation (*driverDistractionSupport* = "false").

Based on the information from the *driverDistractionSupport* element, the MirrorLink Server may provide differentiated services. The value is provided to applications via the Common API. An application may use this information to support various features in a vehicle, i.e. full mirroring, launching the mobile device centric application, video streaming, or it need not support both Drive and Park mode.

Note, that the *driverDisractionSupport* element is supposed have a constant value during a MirrorLink session, i.e. the element shall not replace the VNC *Device Status Drive Mode* flag.

The MirrorLink Client should provide information about its *localization* support with respect to the support of foreign language character sets. In case the information is provided, the MirrorLink Client shall include all supported character sets, as defined by the UniCode Character Code Chart given by the provided entry point, specified in [6].

NOTE: The Unicode code charts define a range for the respective code. The entry point is defined as the first value within that given range. E.g. Basic Latin (ASCII) has a range of 0x0000 - 0x007F. Therefore its entry point is 0x0000.

A MirrorLink Client shall support all characters from a listed Code Chart. A MirrorLink Client shall support at least Basic Latin (ASCII), which is defined by the Character Code Chart entry 0x000.

The MirrorLink Client should not send UniCode key events to the MirrorLink Server, which are not supported from the MirrorLink Server device.

The following applies from MirrorLink 1.3 onwards.

MirrorLink 1.1 and 1.2 Clients may include a *contentRules* element (and sub-elements), at a position between the *rtpStreaming* and the *services* element. The MirrorLink Server should ignore this element entirely.

The MirrorLink Client shall provide information within the *mlUiMode* element about its supported MirrorLink Modes, as defined in [4]. The MirrorLink Client shall include all supported modes:

- `"immersive"` - support for Immersive MirrorLink Mode.
- `"classic"` - support for Classic MirrorLink Mode (mandatory for MirrorLink 1.3 Clients).

MirrorLink 1.1 and 1.2 Clients will not provide any MirrorLink Mode information. Those devices implement Legacy MirrorLink Mode. MirrorLink 1.3 Client devices shall support Immersive MirrorLink Mode and Classic MirrorLink Modes. More details are defined in [4].

The MirrorLink Client shall provide information within the *mlUiControl* element about its supported MirrorLink controls. The MirrorLink Client shall include all supported controls:

- `"pointer"` - supports pointer events via the MirrorLink Client display.
- `"touch"` - supports multi-touch events via the MirrorLink Client display.
- `"knob"` - supports a full rotary knob controllers, as defined in [2].
- `"vc_bt_hfp"` - supports voice command via Bluetooth hands-free profile.
- `"vc_rtp"` - supports voice command via RTP.
- `"ptt"` - provides a Push-to-Talk (PTT) button.

MirrorLink 1.1 and 1.2 Clients will not provide any MirrorLink control information; it therefore has to be extracted via other means (e.g. VNC/WFD session information, UPnP application listings).

MirrorLink Clients shall indicate within the *serverInfo* element, which status information it is going to provide via its own user interfaces, independent of the MirrorLink window, e.g. via a secondary instrument cluster displays. The MirrorLink Client is retrieving this information from the MirrorLink Server via separate means, e.g. via Bluetooth HFP. Information, which is indicated from MirrorLink Clients to be presented, shall not be shown from MirrorLink Servers, as part of a status or navigation bar, to prevent consumer confusion by possible divergent information (this applies to both Immersive and Classic MirrorLink Mode). The MirrorLink Client shall include all shown status information:

- `"battery"` - Battery level of the MirrorLink Server device.
- `"time"` - Local time
- `"operator"` - Network operator.
- `"signalStrength"` - Network signal strength.

In case no information is shown, the MirrorLink Client shall use `"none"`; in this case, no other status information shall be included. MirrorLink 1.1 and 1.2 Clients will not provide any status information, which a MirrorLink Server shall treat as `"none"`. In case the MirrorLink Client loses access to the information, or gains access to the information, it shall send an updated Client Profile to the MirrorLink Server.

The MirrorLink Client shall include a *clientDevice* and a *clientCA* certificates into the client profile. These certificates are provided from the Car Connectivity Consortium as part of the MirrorLink device certification. They state, that the MirrorLink Client device has been certified. The MirrorLink Server shall validate the devices using the Certificate Authority's root certificate as the trust root. Details are defined in clause 5.2.

The MirrorLink Client shall validate the received *Signature*. A failure to successfully validate the *Signature* shall terminate the MirrorLink session.

NOTE: The public key needed to validate the received *Signature* is provided through the Device Attestation Protocol, bound to the *TerminalMode:UPnP-Server* component [7]. Therefore, the MirrorLink Client will either store (parts of) the received Client Profile or retrieve it again, using a *SetClientProfile* or *GetClientProfile* action.

4.2.4 A_ARG_TYPE_ProfileID

An unsigned 32-bit integer representing a unique profile identifier. Its value is set equal to 0 by default.

4.2.5 A_ARG_TYPE_String

A simple string type (UTF-8).

4.2.6 A_ARG_TYPE_INT

An unsigned 32-bit integer represented in decimal (base 10) format.

4.2.7 A_ARG_TYPE_Bool

A simple Boolean string which can either have the value `"true"` or `"false"`.

4.2.8 MaxNumProfiles

An unsigned 16-bit integer greater than or equal to 1, whose value is equal to the maximum number of profiles allowed by *TmClientProfile* service. The value of this state variable is set by the *TmClientProfile* service when it starts up and remains static throughout the lifetime of the service.

Support for multiple Client Profiles reserved for future use, therefore *MaxNumProfiles* shall be set to 1.

4.3 Eventing and Moderation

Table 4-3: Eventing and Moderation

Variable Name	Evented	Moderated Event	Max. Event Rate	Logical Relation	Min. Delta per Event
UnusedProfileIDs	Yes	N/A	NA	N/A	N/A
A_ARG_TYPE_ClientProfile	No	N/A	N/A	N/A	N/A
A_ARG_TYPE_ProfileID	No	N/A	N/A	N/A	N/A
A_ARG_TYPE_String	No	N/A	N/A	N/A	N/A
A_ARG_TYPE_INT	No	N/A	N/A	N/A	N/A
A_ARG_TYPE_Bool	No	N/A	N/A	N/A	N/A
MaxNumProfiles	No	N/A	N/A	N/A	N/A

4.4 Managing Multiple Client Profiles

Support for multiple Client Profiles is reserved for future use.

4.5 Actions

4.5.1 General

Table 4-4: Actions

Name	Device R/O (see note 1)	Control Point R/O (see note 2)
GetMaxNumProfiles	R	O
SetClientProfile	R	R
GetClientProfile	R	O
NOTE 1: For a device this column indicates whether the action needs to 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). NOTE 2: For a control point this column indicates whether a control point needs to 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).		

Implementation Note:

MirrorLink 1.1 Clients may not support *SetClientProfile* action. In this case, the MirrorLink Server has to use defined default values.

4.5.2 GetMaxNumProfiles

4.5.2.1 General

The *GetMaxNumProfiles* action returns the value of the state variable *MaxNumProfiles*, which denotes the maximum number of client profiles that are supported simultaneously by the *TmClientProfile* service.

4.5.2.2 Arguments

Table 4-5: Arguments for GetMaxNumProfiles

Argument	Direction	relatedStateVariable
NumProfilesAllowed	OUT	MaxNumProfiles

Argument:

None.

Return Value:

NumProfilesAllowed (MaxNumProfiles) - Maximum number of client profiles supported.

4.5.2.3 Error Codes for GetMaxNumProfiles

Table 4-6: Error Codes for GetMaxNumProfiles

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture section on Control.
500-599	TBD	See UPnP Device Architecture section on Control.
600-699	TBD	See UPnP Device Architecture section on Control.
701	Operation Rejected	The TmClientProfile service has rejected the operation.
815	Device Locked	The action cannot be processed as the device hosting the TmClientProfileService is locked. User needs to unlock the device first.

4.5.3 SetClientProfile

4.5.3.1 General

The *SetClientProfile* action allows the control point to register a client profile and notify the MirrorLink device about its preferences, settings and capabilities. The MirrorLink UPnP Control Point should invoke the *SetClientProfile* action to add or modify a client profile.

In case the MirrorLink UPnP Control Point does not support the *TmClientProfile* service, the parameters described in the ClientProfile state variable will be assigned their default values.

4.5.3.2 Arguments

Table 4-7: Arguments for SetClientProfile

Argument	Direction	relatedStateVariable
ProfileID	IN	A_ARG_TYPE_ProfileID
ClientProfile	IN	A_ARG_TYPE_ClientProfile
ResultProfile	OUT	A_ARG_TYPE_ClientProfile

Argument:

ProfileID (A_ARG_TYPE_ProfileID) - *ProfileID* of client profile. Reserved for future. Shall be set to `"0"`.

ClientProfile (A_ARG_TYPE_ClientProfile) - Profile information about MirrorLink Client and its capabilities which needs to be updated in the profile record.

In case the *clientProfile* input argument has its value set equal to an empty string, then the *TmClientProfile* shall reset all parameter values for the profile identified by *profileID*, to their default values. Additionally, the MirrorLink Client terminates the MirrorLink session.

Return Value:

ResultProfile (A_ARG_TYPE_ClientProfile) - The updated client profile.

This action can be invoked multiple times with different sets of optional parameters. Multiple invocations of this action will result in the update of client profile settings. However, only those parameters which are specified within the *ClientProfile* input argument of the action, will be updated. The MirrorLink Client shall not update any of the following parameters, while a MirrorLink session is ongoing:

- `clientID`
- `friendlyName`
- `manufacturer`
- `modelName`
- `modelNumber`

If the MirrorLink Server does not support one of the following parameter setting, it shall set the particular value to its default value:

- `rtpStreaming/audioIPL`
- `rtpStreaming/audioMPL`

In case the MirrorLink Client, incorrectly, provides a parameter setting greater than the allowed maximum value, the MirrorLink Server shall cap it at the defined maximum and shall return the capped value in the *SetClientProfile* response. No error code shall be returned. This shall apply for the following parameter:

- `rtpStreaming/lssMax`
- `rtpStreaming/lssAvg`

The MirrorLink Server shall not change any other parameter setting. The MirrorLink UPnP Control Point can utilize the return value to determine which parameter settings in the profile were updated and which parameter settings remained unchanged.

Implementation Note:

> The MirrorLink Server can exclude optional parameters from the *ClientProfile* for non-supported features, even if those parameters had been included into the *ClientProfile* from the MirrorLink Client. E.g. a MirrorLink Server can exclude the *wifi* element, in case it does not support MirrorLink over Wi-Fi.

4.5.3.3 Error Codes for SetClientProfile

Table 4-8: Error Codes for SetClientProfile

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture section on Control.
500-599	TBD	See UPnP Device Architecture section on Control.
600-699	TBD	See UPnP Device Architecture section on Control.
701	Operation Rejected	The TmClientProfile service has rejected the operation.
814	Resource Busy	The requested profile resource is busy. This error occurs when the specific client profile is being accessed and used by another MirrorLink service and any modifications to profile parameters at this moment might cause conflict issues.
815	Device Locked	The action cannot be processed as the device hosting the TmClientProfileService is locked. User needs to unlock the device first.
825	Invalid Profile	The clientProfile argument passed does not conform to A_ARG_TYPE_ClientProfile XML specifications.

ErrorCode	errorDescription	Description
830	Invalid Profile ID	The profile identifier is either invalid or does not exist.

4.5.4 GetClientProfile

4.5.4.1 General

The *GetClientProfile* action allows the control point to access the contents of a client profile stored in the MirrorLink device. On invocation of this action, the *TmClientProfile* service shall use the *profileID* passed as input argument to access the profile information stored on the MirrorLink device, and return it.

The *GetClientProfile* action shall not be executed prior setting the initial profile using the *SetClientProfile* action.

A MirrorLink 1.0, 1.1 or 1.2 Client may execute *GetClientProfile* action prior setting the initial profile using the *SetClientProfile*. In this case, if a profile has never been updated by any MirrorLink UPnP Control Point using the *SetClientProfile* action, then invocation of the *GetClientProfile* action using its *profileID* shall return the profile populated with default parameter values.

4.5.4.2 Arguments

Table 4-9: Arguments for GetClientProfile

Argument	Direction	relatedStateVariable
ProfileID	IN	A_ARG_TYPE_ProfileID
ClientProfile	OUT	A_ARG_TYPE_ClientProfile

Argument:

ProfileID (A_ARG_TYPE_ProfileID) - ProfileID of client profile. Reserved for future. Shall be set to "0".

Return Value:

ClientProfile (A_ARG_TYPE_ClientProfile) - Client profile corresponding to the *profileID* input variable.

4.5.4.3 Error Codes for GetClientProfile

Table 4-10: Error Codes for GetClientProfile

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture section on Control.
500-599	TBD	See UPnP Device Architecture section on Control.
600-699	TBD	See UPnP Device Architecture section on Control.
701	Operation Rejected	The TmClientProfile service has rejected the operation.
815	Device Locked	The action cannot be processed as the device hosting the TmClientProfileService is locked. User needs to unlock the device first.
830	Invalid Profile ID	The profile identifier is either invalid or does not exist.

4.5.5 Relationships Between Actions

None.

4.5.6 Error Code Summary

Table 4-11 lists error codes common to actions for this service type. If an action results in multiple errors, the most specific error should be returned.

Table 4-11: Error Code Summary

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture section on Control.
500-599	TBD	See UPnP Device Architecture section on Control.
600-699	TBD	See UPnP Device Architecture section on Control.
701	Operation Rejected	The *TmClientProfile* service has rejected the operation. MirrorLink Client should retry the action.
814	Resource Busy	The requested profile resource is busy, This error occurs when the specific client profile is being accessed and used by another MirrorLink service and any modifications to profile parameters at this moment might cause conflict issues. MirrorLink Client should retry the action.
815	Device Locked	The action cannot be processed as the device hosting the *TmClientProfileService* is locked. User needs to unlock the device first. MirrorLink Client should not retry the action.
825	Invalid Profile	The *clientProfile* argument passed does not conform to *A_ARG_TYPE_ClientProfile* XML specifications. The MirrorLink Client should verify the format of the argument. MirrorLink Client should not retry the action with the same argument.
830	Invalid Profile ID	The profile identifier is either invalid or does not exist. MirrorLink Client should check the client profile (*GetClientProfile*) and its application support from the *GetApplicationList* response, and retry the action. MirrorLink Client should not retry the action with the same arguments.

NOTE: 800-899 Error Codes are not permitted for standard actions. See UPnP Device Architecture section on Control for more details.

5 Theory of Operation

5.1 Use of Quotation Marks

Throughout the specification, two kinds of quotation marks may be used:

- Quotation marks as in “`words`” are used to highlight a textual element, for readability purpose only. The quotation marks shall not be used within XML schemata, or within arguments of SOAP actions.
- Quotation marks as in `"music"` are part of the XML or SOAP syntax and shall be maintained.

Example:

If *protocolId* is "VNC", then:

- <protocolId>VNC</protocolId> is valid XML; and
- <protocolId>"VNC"</protocolId> is invalid XML.

5.2 Client Certificates

5.2.1 Client Device Certificate

The MirrorLink Client device certificate shall be a public key X.509 version 3 certificate.

The Client Device certificate shall be signed by the Client CA.

The Client Device certificate should have an expiration date of 30 years from the date of signing, but it shall not be longer than the expiration date of the signing Client CA certificate.

Any Client Device Certificate shall use 2048-bit RSA keys with SHA-256 or SHA-512 signature algorithms.

The X.509 extension header shall have the following format. The CCC-MirrorLink Extension Id is provided from IANA. The identifier shall be provided without any "<>" delimiter. Its value is outside the scope of the present document:

```
X509v3 extensions:
    CCC-MirrorLink-Client-Device Extension:
        extnId:    1.3.6.1.4.1.41577.4.1
        critical:  no
        extnValue: DER:<DER encoded XML, as specified below>
```

The DER encoded XML shall follow the format below.

Table 5-1

Element	Description	Parent	Availability
certification	MirrorLink Application Certificate	-	Required
manufacturer	Manufacturer Name, as presented in the Client Profile. (A_ARG_TYPE_String)	certification	Mandatory
modelNumber	Model Number, as presented in the Client Profile. (A_ARG_TYPE_String)	certification	Mandatory

The MirrorLink Server shall follow the steps given below:

- Validate the application certificate (i.e. check expiration dates, format, and signature).
- Validate the certificate's trust chain.
- Validate the *manufacturer* and *modelNumber* entries (shall be identical to the values provided in the Client Profile).

In any of the above steps fail, the MirrorLink Client is considered non-conforming. The MirrorLink Server shall immediately terminate the MirrorLink session, sending a *SSDP:byebye* message, and terminating all established MirrorLink related connections (e.g. WFD, VNC, RTP, CDB).

The MirrorLink Server shall not terminate the MirrorLink session with MirrorLink 1.0, 1.1 and 1.2 Clients, because of the missing MirrorLink Client certificates. But, a MirrorLink Server shall terminate the MirrorLink session if such a MirrorLink Client starts using MirrorLink 1.3 features.

5.2.2 Client CA Certificate

The MirrorLink Client CA certificate shall be a public key X.509 version 3 certificate.

The Client CA certificate shall be signed by the Certificate Authority, which shall be identical to the DAP root CA.

The Client CA certificate should have an expiration date of 30 years from the date of signing, but it shall not be longer than the expiration date of the signing root certificate.

Any Client CA certificate shall use 4096-bit RSA keys with SHA-512 signature algorithms.

5.2.3 Testing Considerations

For the purposes of testing, MirrorLink Client manufacturers shall use a test root certificate to validate the Client device and Client CA certificates. These certificates shall be used only during device certification of the MirrorLink Client.

For the purposes of Interoperability testing with production devices, MirrorLink Client manufacturers can additionally receive short-lived Client device and Client CA certificates from the Certificate Authority with an expiration date of not more than 6 months from the date of signing. These certificates are used during device certification of the MirrorLink Client.

After successful device certification, the MirrorLink Client manufacturer will receive the final long-lived certificates from the Certificate Authority. Updating the test certificates with these certificates will not trigger an Engineering Change Order (ECO). The Device Certification body may require smoke testing though, to ensure, that the update has not broken any MirrorLink Client functionality.

5.3 Example Values of State Variables

5.3.1 UnusedProfileIDs

The value of *UnusedProfileIDs* state variable is a comma separated list of profile IDs for profiles which are currently not being used by any MirrorLink service hosted on the MirrorLink device. Each entry in the list is of type A_ARG_TYPE_ProfileID.

EXAMPLE: If the value of *MaxNumProfiles* is equal to 5 then the value of *UnusedProfileIDs* when no profile is being used by any service is:

- 0,1,2,3,4

5.3.2 A_ARG_TYPE_ClientProfile

The value of *A_ARG_TYPE_ClientProfile* is an XML block corresponding to a list of preferences, settings and capabilities of the MirrorLink Client.

Example:

The following example illustrates the usage of this variable:

```
<?xml version="1.0" encoding="UTF-8"?>
<clientProfile>
    <clientID>Cl_1</clientID>
    <friendlyName>Client One</friendlyName>
    <manufacturer>man_2</manufacturer>
    <modelName>CL_Model2</modelName>
    <modelNumber>2009</modelNumber>
    <iconPreference>
        <mimetype>image/png</mimetype>
        <width>240</width>
        <height>240</height>
        <depth>24</depth>
    </iconPreference>
    <connectivity>
        <bluetooth>
            <bdAddr>1A2B3C4D5E6F</bdAddr>
            <startConnection>false</startConnection>
```

```
        </bluetooth>
    </connectivity>
    <rtpStreaming>
        <payloadType>0,99</payloadType>
        <audioIPL>4800</audioIPL>
        <audioMPL>9600</audioMPL>
        <lssMax>1250</lssMax>
        <lssAvg>850</lssAvg>
    </rtpStreaming>
    <services>
        <notification>
            <notiUiSupport>true</notiUiSupport>
            <maxActions>3</maxActions>
            <actionNameMaxLength>15</actionNameMaxLength>
            <notiTitleMaxLength>25</notiTitleMaxLength>
            <notiBodyMaxLength>100</notiBodyMaxLength>
        </notification>
    </services>
    <mirrorLinkVersion>
        <majorVersion>1</majorVersion>
        <minorVersion>3</minorVersion>
    </mirrorLinkVersion>
    <presentations>
        <presentation>vncu</presentation>
    </presentations>
    <misc>
        <driverDistractionSupport>true</driverDistractionSupport>
        <mlUiMode>
            <mode>classic</mode>
            <mode>immersive</mode>
        </mlUiMode>
        <mlUiControl>
            <control>pointer</control>
            <control>vc_rtp</control>
            <control>ptt</control>
        </mlUiControl>
        <serverInfo>
            <info>none</info>
        </serverInfo>
    </misc>
    <certificates>
        <clientDevice>[certificate Data]</clientDevice>
        <clientCA>[certificate Data]</clientCA>
    </certificates>
</clientProfile>
```

6 A_ARG_TYPE_ClientProfile XSD Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="urn:schemas-upnp-org:tmclientprofile:clientprofile-1-0"
 xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 elementFormDefault="qualified" attributeFormDefault="unqualified" id="clientprofile">
<xs:import namespace="http://www.w3.org/2000/09/xmldsig#"
 schemaLocation="http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/xmldsig-core-schema.xsd"/>
<xs:element name="clientProfile">
 <xs:complexType>
  <xs:sequence minOccurs="1" maxOccurs="1">
   <xs:element name="clientID" minOccurs="1" maxOccurs="1"
   type="xs:string"/>
   <xs:element name="friendlyName" type="xs:string" minOccurs="0"/>
   <xs:element name="manufacturer" type="xs:string" minOccurs="1"/>
   <xs:element name="modelName" type="xs:string" minOccurs="0"/>
   <xs:element name="modelNumber" type="xs:string" minOccurs="1"/>
   <xs:element name="iconPreference" minOccurs="0">
    <xs:complexType>
     <xs:sequence>
      <xs:element name="mimetype" type="xs:string" minOccurs="0" default="image/png"/>
      <xs:element name="width" type="xs:positiveInteger" minOccurs="0" default="128"/>
      <xs:element name="height" type="xs:positiveInteger" minOccurs="0" default="128"/>
      <xs:element name="depth" type="xs:positiveInteger" minOccurs="0" default="24"/>
      <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
     </xs:sequence>
     <xs:anyAttribute namespace="##any" processContents="lax"/>
    </xs:complexType>
```

```
</xs:element>
<xs:element name="connectivity" minOccurs="0">
 <xs:complexType>
  <xs:sequence>
   <xs:element name="bluetooth" minOccurs="0" maxOccurs="1">
    <xs:complexType>
     <xs:sequence>
      <xs:element name="bdAddr" type="xs:string" minOccurs="0"/>
      <xs:element name="startConnection" type="xs:boolean"
      minOccurs="0" default="true"/>
      <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
     </xs:sequence>
     <xs:anyAttribute namespace="##any" processContents="lax"/>
    </xs:complexType>
   </xs:element>
   <xs:element name="wifi" minOccurs="0" maxOccurs="1">
    <xs:complexType>
     <xs:sequence>
      <xs:element name="macAddr" type="xs:string" minOccurs="1"/>
      <xs:element name="ssid" type="xs:string" minOccurs="0"/>
      <xs:element name="roles" type="xs:string" minOccurs="0" default="AP,Client,P2P"/>
      <xs:element name="protectionList" minOccurs="0" maxOccurs="1">
       <xs:complexType>
        <xs:sequence>
         <xs:element name="protection" minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
           <xs:sequence>
            <xs:element name="protocol" minOccurs="1">
             <xs:simpleType>
              <xs:restriction base="xs:string">
               <xs:enumeration value="WEP"/>
               <xs:enumeration value="WPA"/>
               <xs:enumeration value="WPA2"/>
               <xs:enumeration value="WPS"/>
              </xs:restriction>
             </xs:simpleType>
            </xs:element>
            <xs:element name="passkey" type="xs:string" minOccurs="1"/>
           </xs:sequence>
           <xs:anyAttribute namespace="##any" processContents="lax"/>
          </xs:complexType>
         </xs:element>
        </xs:sequence>
        <xs:anyAttribute namespace="##any" processContents="lax"/>
       </xs:complexType>
      </xs:element>
     </xs:sequence>
     <xs:anyAttribute namespace="##any" processContents="lax"/>
    </xs:complexType>
   </xs:element>
   <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##any" processContents="lax"/>
 </xs:complexType>
</xs:element>
<xs:element name="rtpStreaming" minOccurs="1">
 <xs:complexType>
  <xs:sequence>
   <xs:element name="payloadType" type="xs:string" minOccurs="0" default="99"/>
   <xs:element name="audioIPL" type="xs:positiveInteger" minOccurs="0" default="4800"/>
   <xs:element name="audioMPL" type="xs:positiveInteger" minOccurs="0" default="9600"/>
   <xs:element name="lssMax" type="xs:positiveInteger"/>
   <xs:element name="lssAvg" type="xs:positiveInteger"/>
   <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##any" processContents="lax"/>
 </xs:complexType>
</xs:element>
<xs:element name="contentRules" minOccurs="0">
 <xs:complexType>
  <xs:sequence>
   <xs:element name="rule" minOccurs="0" maxOccurs="unbounded">
    <xs:complexType>
     <xs:sequence>
      <xs:element name="ruleId" type="xs:nonNegativeInteger" minOccurs="1"/>
      <xs:element name="ruleValue" type="xs:string" minOccurs="0"/>
     </xs:sequence>
    </xs:complexType>
```

```
   </xs:element>
  </xs:sequence>
 </xs:complexType>
</xs:element>
<xs:element name="services" minOccurs="0">
 <xs:complexType>
  <xs:sequence>
   <xs:element name="notification" minOccurs="0">
    <xs:complexType>
     <xs:sequence>
      <xs:element name="notiUiSupport" type="xs:boolean" minOccurs="0" default="false"/>
      <xs:element name="maxActions" type="xs:positiveInteger" minOccurs="0" default="2"/>
      <xs:element name="actionNameMaxLength" type="xs:positiveInteger" minOccurs="0"
       default="10"/>
      <xs:element name="notiTitleMaxLength" type="xs:positiveInteger" minOccurs="0"
       default="20"/>
      <xs:element name="notiBodyMaxLength" type="xs:positiveInteger" minOccurs="0" default="80"/>
      <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
     </xs:sequence>
     <xs:anyAttribute namespace="##any" processContents="lax"/>
    </xs:complexType>
   </xs:element>
   <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##any" processContents="lax"/>
 </xs:complexType>
</xs:element>
<xs:element name="mirrorLinkVersion" minOccurs="0" maxOccurs="1">
 <xs:complexType>
  <xs:sequence>
   <xs:element name="majorVersion" type="xs:nonNegativeInteger"/>
   <xs:element name="minorVersion" type="xs:nonNegativeInteger"/>
   <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
  </xs:sequence>
 </xs:complexType>
</xs:element>
<xs:element name="presentations" minOccurs="1">
 <xs:complexType>
  <xs:sequence>
   <xs:element name="presentation" type="xs:string" minOccurs="1"/>
   <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##any" processContents="lax"/>
 </xs:complexType>
</xs:element>
<xs:element name="misc" minOccurs="1">
 <xs:complexType>
  <xs:sequence>
   <xs:element name="driverDistractionSupport" type="xs:boolean" default="true"/>
   <xs:element name="mlUiMode" minOccurs="1" maxOccurs="1">
    <xs:complexType>
     <xs:sequence>
      <xs:element name="mode" minOccurs="1" maxOccurs="unbounded">
       <xs:simpleType>
        <xs:restriction base="xs:string">
         <xs:enumeration value="immersive"/>
         <xs:enumeration value="classic"/>
        </xs:restriction>
       </xs:simpleType>
      </xs:element>
     </xs:sequence>
     <xs:anyAttribute namespace="##any" processContents="lax"/>
    </xs:complexType>
   </xs:element>
   <xs:element name="mlUiControl" minOccurs="1" maxOccurs="1">
    <xs:complexType>
     <xs:sequence>
      <xs:element name="control" minOccurs="1" maxOccurs="unbounded">
       <xs:simpleType>
        <xs:restriction base="xs:string">
         <xs:enumeration value="pointer"/>
         <xs:enumeration value="touch"/>
         <xs:enumeration value="knob"/>
         <xs:enumeration value="vc_bt_hfp"/>
         <xs:enumeration value="vc_rtp"/>
         <xs:enumeration value="ptt"/>
        </xs:restriction>
       </xs:simpleType>
```

```
        </xs:element>
       </xs:sequence>
       <xs:anyAttribute namespace="##any" processContents="lax"/>
      </xs:complexType>
     </xs:element>
     <xs:element name="serverInfo" minOccurs="1" maxOccurs="1">
      <xs:complexType>
       <xs:sequence>
        <xs:element name="info" minOccurs="1" maxOccurs="unbounded">
         <xs:simpleType>
          <xs:restriction base="xs:string">
           <xs:enumeration value="battery"/>
           <xs:enumeration value="time"/>
           <xs:enumeration value="operator"/>
           <xs:enumeration value="signalStrength"/>
           <xs:enumeration value="none"/>
          </xs:restriction>
         </xs:simpleType>
        </xs:element>
       </xs:sequence>
       <xs:anyAttribute namespace="##any" processContents="lax"/>
      </xs:complexType>
     </xs:element>
     <xs:any namespace="##any" minOccurs="0"
      maxOccurs="unbounded" processContents="lax"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
   </xs:complexType>
  </xs:element>
  <xs:element name="localization" minOccurs="0">
   <xs:complexType>
    <xs:sequence>
     <xs:element name="characterSet" type="xs:string"/>
     <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
   </xs:complexType>
  </xs:element>
  <xs:element name="certificates" minOccurs="0">
   <xs:complexType>
    <xs:sequence>
     <xs:element name="clientDevice" type="xs:string" minOccurs="0"/>
     <xs:element name="clientCA" type="xs:string" minOccurs="0"/>
     <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
   </xs:complexType>
  </xs:element>
  <xs:element ref="ds:Signature" minOccurs="1"/>
  <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
 </xs:sequence>
 <xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:complexType>
</xs:element>
</xs:schema>
```

7 XML Service Description

```
<?xml version="1.0" encoding="UTF-8"?>
<scpd xmlns="urn:schemas-upnp-org:service-1-0">
    <specVersion>
        <major>1</major>
        <minor>0</minor>
    </specVersion>
    <actionList>
        <action>
            <name>GetMaxNumProfiles</name>
            <argumentList>
                <argument>
                    <name>NumProfilesAllowed</name>
                    <direction>out</direction>
                    <relatedStateVariable>
                        MaxNumProfiles
                    </relatedStateVariable>
                </argument>
```

```
            </argumentList>
        </action>
        <action>
            <name>SetClientProfile</name>
            <argumentList>
                <argument>
                    <name>ProfileID</name>
                    <direction>in</direction>
                    <relatedStateVariable>
                        A_ARG_TYPE_ProfileID
                    </relatedStateVariable>
                </argument>
                <argument>
                    <name>ClientProfile</name>
                    <direction>in</direction>
                    <relatedStateVariable>
                        A_ARG_TYPE_ClientProfile
                    </relatedStateVariable>
                </argument>
                <argument>
                    <name>ResultProfile</name>
                    <direction>out</direction>
                    <relatedStateVariable>
                        A_ARG_TYPE_ClientProfile
                    </relatedStateVariable>
                </argument>
            </argumentList>
        </action>
        <action>
            <name>GetClientProfile</name>
            <argumentList>
                <argument>
                    <name>ProfileID</name>
                    <direction>in</direction>
                    <relatedStateVariable>
                        A_ARG_TYPE_ProfileID
                    </relatedStateVariable>
                </argument>
                <argument>
                    <name>ClientProfile</name>
                    <direction>out</direction>
                    <relatedStateVariable>
                        A_ARG_TYPE_ClientProfile
                    </relatedStateVariable>
                </argument>
            </argumentList>
        </action>
    </actionList>
    <serviceStateTable>
        <stateVariable sendEvents="yes">
            <name>UnusedProfileIDs</name>
            <dataType>string</dataType>
        </stateVariable>
        <stateVariable sendEvents="no">
            <name>A_ARG_TYPE_ClientProfile</name>
            <dataType>string</dataType>
        </stateVariable>
        <stateVariable sendEvents="no">
            <name>A_ARG_TYPE_ProfileID</name>
            <dataType>ui4</dataType>
            <defaultValue>0</defaultValue>
        </stateVariable>
        <stateVariable sendEvents="no">
            <name>A_ARG_TYPE_String</name>
            <dataType>string</dataType>
        </stateVariable>
        <stateVariable sendEvents="no">
            <name>A_ARG_TYPE_INT</name>
            <dataType>ui4</dataType>
        </stateVariable>
        <stateVariable sendEvents="no">
            <name>A_ARG_TYPE_Bool</name>
            <dataType>string</dataType>
        </stateVariable>
        <stateVariable sendEvents="no">
            <name>MaxNumProfiles</name>
            <dataType>ui2</dataType>
            <defaultValue>1</defaultValue>
```

```
        </stateVariable>
    </serviceStateTable>
</scpd>
```

Annex A (informative): Authors and Contributors

The following people have contributed to the present document:

Rapporteur:	Dr. Jörg Brakensiek, E-Qualus (for Car Connectivity Consortium LLC)
Other contributors:	Raja Bose, Nokia Corporation

History

Document history		
V1.3.0	October 2017	Publication
V1.3.1	October 2019	Publication