Car Connectivity Consortium

MirrorLink®

UPnP Client Profile Service

Version 1.1.4
(CCC-TS-026)

Version History

Version	Date	Comment
1.0.1	26 June 2011	Approved Version
1.1	31 March 2012	Approved Version
1.1.1	16 October 2012	Approved Errata Version
1.1.2	20 December 2012	Approved Errata Version
1.1.3	05 March 2013	Approved Errata Version
1.1.4	05 November 2013	Approved Errata Version

Contributors

Bose, Raja,	Nokia Corporation
Brakensiek, Jörg (Editor)	Nokia Corporation

Trademarks

MirrorLink is a registered trademark of Car Connectivity Consortium LLC

Bluetooth is a registered trademark of Bluetooth SIG Inc.

RFB and VNC are registered trademarks of RealVNC Ltd.

UPnP is a registered trademark of UPnP Forum.

Other names or abbreviations used in this document may be trademarks of their respective owners.

Legal Notice

The copyright in this Specification is owned by the Car Connectivity Consortium LLC ("CCC LLC"). Use of this Specification and any related intellectual property (collectively, the "Specification"), is governed by these license terms and the CCC LLC Limited Liability Company Agreement (the "Agreement").

Use of the Specification by anyone who is not a member of CCC LLC (each such person or party, a "Member") is prohibited. The legal rights and obligations of each Member are governed by the Agreement and their applicable Membership Agreement, including without limitation those contained in Article 10 of the LLC Agreement.

CCC LLC hereby grants each Member a right to use and to make verbatim copies of the Specification for the purposes of implementing the technologies specified in the Specification to their products ("Implementing Products") under the terms of the Agreement (the "Purpose"). Members are not permitted to make available or distribute this Specification or any copies thereof to non-Members other than to their Affiliates (as defined in the Agreement) and subcontractors but only to the extent that such Affiliates and subcontractors have a need to know for carrying out the Purpose and provided that such Affiliates and subcontractors accept confidentiality obligations similar to those contained in the Agreement. Each Member shall be responsible for the observance and proper performance by such of its Affiliates and subcontractors of the terms and conditions of this Legal Notice and the Agreement. No other license, express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein.

Any use of the Specification not in compliance with the terms of this Legal Notice, the Agreement and Membership Agreement is prohibited and any such prohibited use may result in termination of the applicable Membership Agreement and other liability permitted by the applicable Agreement or by applicable law to CCC LLC or any of its members for patent, copyright and/or trademark infringement.

THE SPECIFICATION IS PROVIDED "AS IS" WITH NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION ANY WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS, AND COMPLIANCE WITH APPLICABLE LAWS.

Each Member hereby acknowledges that its Implementing Products may be subject to various regulatory controls under the laws and regulations of various jurisdictions worldwide. Such laws and regulatory controls may govern, among other things, the combination, operation, use, implementation and distribution of Implementing Products. Examples of such laws and regulatory controls include, but are not limited to, road safety regulations, telecommunications regulations, technology transfer controls and health and safety regulations. Each Member is solely responsible for the compliance by their Implementing Products with any such laws and regulations and for obtaining any and all required authorizations, permits, or licenses for their Implementing Products related to such regulations within the applicable jurisdictions.

Each Member acknowledges that nothing in the Specification provides any information or assistance in connection with securing such compliance, authorizations or licenses**.**

NOTHING IN THE SPECIFICATION CREATES ANY WARRANTIES, EITHER EXPRESS OR IMPLIED, REGARDING SUCH LAWS OR REGULATIONS. ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF ANY INTELLECTUAL PROPERTYRIGHTS OR FOR NONCOMPLIANCE WITH LAWS, RELATING TO USE OF THE SPECIFICATION IS EXPRESSLY DISCLAIMED. BY USE OF THE SPECIFICATION, EACH MEMBER EXPRESSLY WAIVES ANY CLAIM AGAINST CCC LLC AND ITS MEMBERS RELATED TO USE OF THE SPECIFICATION.

CCC LLC reserve the right to adopt any changes or alterations to the Specification as it deems necessary or appropriate.

Contents

List of Tables

1 Overview and Scope

This service definition is compliant with the UPnP Device Architecture version 1.0 [1]. It defines a service type referred to herein as TmClientProfile service.

The specification lists a series of requirements, either explicitly or within the text, which are mandatory elements for compliant solutions. Recommendations are given, to ensure optimal usage and to provide suitable performance. All recommendations are optional.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document follow the notation as described in RFC 2119 [1].

- MUST: This word, or the terms "REQUIRED" or "SHALL", means that the definition is an absolute requirement of the specification.
- MUST NOT: This phrase, or the phrase "SHALL NOT", means that the definition is an absolute prohibition of the specification.
- SHOULD: This word, or the adjective "RECOMMENDED", means that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
- SHOULD NOT: This phrase, or the phrase "NOT RECOMMENDED" means that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
- MAY: This word, or the adjective "OPTIONAL", means that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option MUST be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option MUST be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides.)

1.1 Introduction

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

2 Service Modeling Definitions

2.1 Service Type

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

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

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

2.2 State Variables

Table 2-1: 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
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

[1] R = *REQUIRED,* O = *OPTIONAL,* X = *Non-standard*

2.2.1 UnusedProfileIDs

A string formatted as UTF-8 which consists of a comma separated list of profile IDs 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 profile IDs changes.

2.2.2 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 the following table.

Table 2-2: Structure of A_ARG_TYPE_ClientProfile

Element	Description	Parent	Availability
clientProfile	Profile Information about a MirrorLink Client	-	Required
clientID	ID of the MirrorLink Client (A_ARG_TYPE_String)	clientProfile	Required
friendlyName	Short user-friendly description of client (A_ARG_TYPE_String)	clientProfile	Optional
manufacturer	Manufacturer Name (A_ARG_TYPE_String)	clientProfile	Optional
modelName	Model Name (A_ARG_TYPE_String)	clientProfile	Optional
modelNumber	Model Number (A_ARG_TYPE_String)	clientProfile	Optional
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

Element	Description	Parent	Availability
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
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	protection	Mandatory

Element	Description	Parent	Availability
	MUST be left empty, if transmitted over an unprotected or shared transport channel (e.g. WiFi) (A_ARG_TYPE_String)		
rtpStreaming	RTP streaming parameters	clientProfile	Optional
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
audioMPL	Audio Maximum Playback Length in reference to payload type 99. (A_ARG_TYPE_INT) **Default**: “9600“	rtpStreaming	Optional
contentRules	Application UI content MUST follow certain UI rules. These rules are implementation-specific and beyond the scope of this document. Depreciated;	clientProfile	Optional
rule*	User Interface property is following a certain UI norm of a given year. Depreciated;	contentRules	Optional
ruleId	Unique rule identifier (ruleID) as defined in the TmApplicationServer:1 service description (Appendix A – Table Content Rules). Allowed values are 0 to 31. Depreciated; MirrorLink Server SHOULD ignore value. (A_ARG_TYPE_INT)	rule	Mandatory
ruleValue	Specific value, which need to be fulfilled Depreciated (A_ARG_TYPE_String)	rule	Optional
services	Profile information on different services supported from the client	clientProfile	Optional
notification	Configuration of the UPnP TmNotificationService	services	Optional
notiUiSupport	Support for native notification U (A_ARG_TYPE_Bool) **Default**: “false”	notification	Optional
maxActions	Maximum number of actions MUST equal or greater than 0. 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 MUST be greater than 0. (A_ARG_TYPE_INT) **Default**: “10”	notification	Optional

Element	Description	Parent	Availability
notiTitle MaxLength	Maximum supported length of the notification title. MUST be greater than 0. (A_ARG_TYPE_INT) **Default**: `"20"`	notification	Optional
notiBody MaxLength	Maximum supported length of the notification body. MUST be greater than 0. (A_ARG_TYPE_INT) **Default**: `"80"`	notification	Optional
mirrorLink Version	MirrorLink Client version	clientProfile	Optional
majorVersion	Major Version A_ARG_TYPE_INT	mirrorLink Version	Mandatory
minorVersion	MinorVersion A_ARG_TYPE_INT	mirrorLink Version	Mandatory

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

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

The MirrorLink Client MUST 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 MUST 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.

2.2.3 A_ARG_TYPE_ProfileID

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

2.2.4 A_ARG_TYPE_String

A simple string type (UTF-8).

2.2.5 A_ARG_TYPE_INT

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

2.2.6 A_ARG_TYPE_Bool

A simple Boolean string which can either have the value ‘true’ or ‘false’.

2.2.7 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.

2.3 Eventing and Moderation

Table 2-3: 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

Variable Name	Evented	Moderated Event	Max. Event Rate	Logical Relation	Min. Delta per Event
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

2.4 Managing Multiple Client Profiles

The TmClientProfile service can support multiple client profiles. This enables multiple control points to utilize the services hosted by the MirrorLink device simultaneously with different client specific settings.

Essentially, a profile directory is provided, where each individual profile is referenced by its ProfileID number. The TmClientProfile service supports a fixed number of entries in the profile directory which can be used by MirrorLink Clients to store custom profiles. This number is stored in the static state variable MaxNumProfiles. The GetMaxNumProfiles action can be used by the MirrorLink control point to determine the maximum number of profile entries in the profile directory (numProfilesAllowed). The different profiles are identified by a Profile Identifier (or ProfileID) which is an integer in the range [0, numProfilesAllowed-1]. For example, if the value of numProfilesAllowed is equal to 5, then the valid ProfileID values are 0, 1, 2, 3 and 4.

Note that the support for multiple client profiles is OPTIONAL. A TmClientProfile service implementation that does not support multiple client profiles SHOULD set the value of numProfilesAllowed equal to 1. This would imply that only one profile is available on the device and has ProfileID equal to 0.

The implementation of the profile directory MUST be located in shared memory accessible to all the services hosted on the MirrorLink device. Each record in the directory MUST be associated with a specific profile and consist of the following elements:

1) ProfileID: The unique identifier of the profile, whose valid range is as described above. The ProfileID serves as the record locator of a profile in the profile directory.

2) Profile: The profile information contains all the parameters associated with the profile and their respective values.

3) Semaphore[1]: This is a counting semaphore which stores the number of services in the MirrorLink device which are accessing and using the profile at any given time. The semaphore has its value initialized equal to 0. Any service which wants to access and use the profile MUST increment the value of the semaphore by 1. Similarly any service which wants to relinquish its access to a profile MUST decrement the value of the semaphore by 1. The purpose of the semaphore is to ensure that there are no synchronization issues or conflicts by enforcing a multi-reader/single-writer policy, where multiple services can use the same profile simultaneously but any modification of the contents of a profile MUST wait till no other service is using it (i.e. when the value of the semaphore is equal to 0).

[1] This specification does not mandate the use of a specific implementation of Semaphore, rather uses the term Semaphore for its concept.

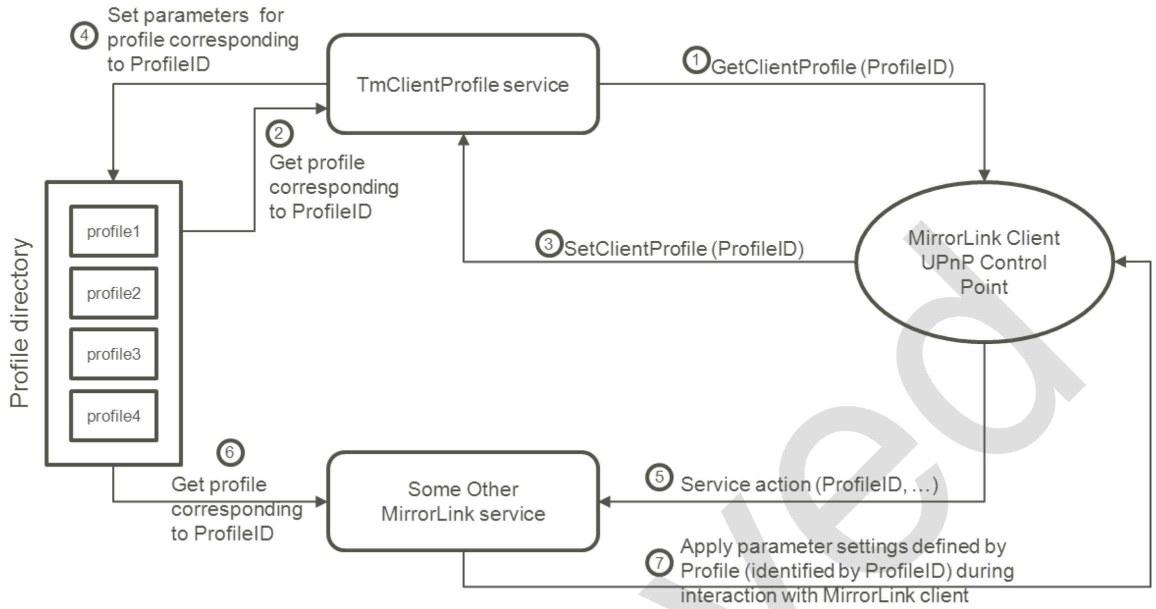
4 Set parameters for profile corresponding to ProfileID
TmClientProfile service
1 GetClientProfile (ProfileID)
2 Get profile corresponding to ProfileID
Profile directory
profile1
profile2
profile3
profile4
3 SetClientProfile (ProfileID)
MirrorLink Client UPnP Control Point
6 Get profile corresponding to ProfileID
Some Other MirrorLink service
5 Service action (ProfileID, …)
7 Apply parameter settings defined by Profile (identified by ProfileID) during interaction with MirrorLink client

Figure 1: Access, Modification and Utilization of Client Profiles by MirrorLink UPnP Control Point

When the TmClientProfile is first started, the profile associated with each ProfileID is populated with default values. MirrorLink control points can utilize the TmClientProfile service to access and modify the profile associated with any of the possible ProfileIDs lying in the range [0, numProfilesAllowed-1], as shown in Figure 1 (steps 1 to 4). After that they can utilize the same ProfileID with other services hosted on the MirrorLink device (for example, TmApplicationServer) to indicate that they would like the specific parameter settings associated with that profile to be used during their interaction with the MirrorLink device, as shown in Figure 1 (steps 5 to 7). In this manner, multiple clients may have their own custom profiles (limited in number by the value of numAllowedProfiles) for governing their interaction with services hosted on the MirrorLink device.

A TmClientProfile service may be prevented from modifying a specific profile in case that profile is already being actively used and accessed by another MirrorLink service. In such a case, an error (814) will be returned.

2.5 Actions

Table 2-4: Actions

Name	Device R/O[1]	Control Point R/O[2]
GetMaxNumProfiles	R	O
SetClientProfile	R	O
GetClientProfile	R	O

[1] 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_).

[2] 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 GetMaxNumProfiles

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.

2.5.1.1 Arguments

Table 2-5: Arguments for GetMaxNumProfiles

Argument	Direction	relatedStateVariable
NumProfilesAllowed	OUT	MaxNumProfiles

Argument:

None.

Return Value:

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

2.5.1.2 Error Codes for GetMaxNumProfiles

Table 2-6: 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.

2.5.2 SetClientProfile

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.

2.5.2.1 Arguments

Table 2-7: 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) – The identifier of the profile record where the client profile settings MUST be stored.

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 MUST reset all parameter values for the profile identified by profileID, to their default values.

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 updation of client profile settings for a specific profile stored in the MirrorLink device. However, only those parameters which are specified within the clientProfile input argument of the action, will be updated. If the MirrorLink Server does not support a parameter setting, it MUST set the particular value to its default value. 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.

2.5.2.2 Error Codes for SetClientProfile

Table 2-8: 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.
830	Invalid Profile ID	The profile identifier is either invalid or does not exist.

2.5.3 GetClientProfile

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 MUST use the profileID passed as input argument to access the profile information stored on the MirrorLink device, and return it.

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 MUST return the profile populated with default parameter values.

2.5.3.1 Arguments

Table 2-9: Arguments for GetClientProfile

Argument	Direction	relatedStateVariable
ProfileID	IN	A_ARG_TYPE_ProfileID
ClientProfile	OUT	A_ARG_TYPE_ClientProfile

Argument:

ProfileID (A_ARG_TYPE_ProfileID) – The identifier of the profile record where the client profile settings MUST be stored.

Return Value:

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

2.5.3.2 Error Codes for GetClientProfile

Table 2-10: 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.

2.5.4 Relationships Between Actions

None.

2.5.5 Error Code Summary

The following table 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 2-11: 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.

ErrorCode	errorDescription	Description
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.

3 Theory of Operation

3.1 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 MUST 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 MUST be maintained.

Example:

If <protocolId> must be `“VNC”`, then

- `<protocolId>VNC</protocolId>` is valid XML and
- `<protocolId>“VNC”</protocolId>` is invalid XML.

3.2 Example Values of State Variables

3.2.1 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

3.2.2 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>
   </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>1</minorVersion>
   </mirrorLinkVersion>
</clientProfile>
```

4 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:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified"
id="clientprofile">
<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="0"/>
   <xs:element name="modelName" type="xs:string" minOccurs="0"/>
   <xs:element name="modelNumber" type="xs:string" minOccurs="0"/>
   <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="0" maxOccurs="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: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:any namespace="##any" minOccurs="0"
          maxOccurs="unbounded" processContents="lax"/>
         </xs:sequence>
         <xs:anyAttribute namespace="##any" processContents="lax"/>
        </xs:complexType>
       </xs:element>
      <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"
      processContents="lax"/>
     </xs:sequence>
     <xs:anyAttribute namespace="##any" processContents="lax"/>
    </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:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
   processContents="lax"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##any" processContents="lax"/>
 </xs:complexType>
</xs:element>
```

```
</xs:schema>
```

5 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>
```

6 References

[1] UPnP Forum, "UPnP Device Architecture 1.0", 24 April 2008, http://www.upnp.org

[2] IETF, RFC 2119, “Keys words for use in RFCs to Indicate Requirement Levels”, March 1997. http://www.ietf.org/rfc/rfc2119.txt

[3] Car Connectivity Consortium, “MirrorLink 1.1 – Handling of Application Certificates”, Version 1.1; CCC-TS-036