Car Connectivity Consortium
MirrorLink®

UPnP Notification Server Service Test Specification

Version 1.1.5
(CCC-TS-029)

VERSION HISTORY

Version	Date	Comment
1.1	31 March 2012	Approved Version
1.1.1	27 September 2012	Approved Errata Version
1.1.2	05 March 2013	Approved Errata Version
1.1.3	05 November 2013	Approved Errata Version
1.1.4	25 September 2014	Approved Errata Version
1.1.5	10 November 2014	Approved Errata Version

LIST OF CONTRIBUTORS

Brakensiek, Jörg (Editor)	Microsoft Corporation
Hrabak, Robert	General Motors Corporation
Lehner, Martin	jambit GmbH

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.

TABLE OF CONTENTS

TERMS AND ABBREVIATIONS

DAP	Device Attestation Protocol
UPnP	Universal Plug and Play

MirrorLink is a trademark of the 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.

1 ABOUT

This document specifies all MirrorLink protocol conformance test cases for the UPnP Notification Server Service specification [2].

The specification lists a series of requirements, either explicitly or within the text, which are mandatory elements for a 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 are following the notation as described in RFC 2119 [1].

1. MUST: This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.
2. MUST NOT: This phrase, or the phrase "SHALL NOT", mean that the definition is an absolute prohibition of the specification.
3. SHOULD: This word, or the adjective "RECOMMENDED", mean 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.
4. SHOULD NOT: This phrase, or the phrase "NOT RECOMMENDED" mean 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.
5. 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.)

2 DEFINITIONS

2.1 Execution of Test Cases

Every test case is uniquely identified by an identifier.

- A MirrorLink server MUST pass all test cases, starting with SR.
- A MirrorLink client MUST pass all test cases, starting with CL

Every test case description includes an entry, whether the test cases is considered mandatory or not.

- Test cases marked as MANDATORY, MUST be executed.
- Test cases marked as CONDITIONAL, MUST be executed if the given condition is met.
- Test cases marked as CONDITIONAL, MUST NOT be executed if the given condition is not met.
- Test cases marked as NONE, MUST NOT be executed

2.2 Server Definitions

2.2.1 UPnP Server Register Notifications

This definition contains all necessary steps to register for notifications from all applications.

Step	Name	Description	Expected Result
1	UPnP Server Connect	See definitions in [3]	
2	UPnP Application Launch	Test Engineer MAY need to launch applications supporting notifications. • Invoke Get Application List action • Invoke Launch application action for application(s) to be started. This step MAY be skipped.	• Provide application list (if needed) • Launch applications (if needed)
3	UPnP Get Supported Applications	Invoke Get Supported Application • ProfileID set to 0	• Receive comma separated list of application ids. • Must be unique • Must not be empty
4	UPnP Set Allowed Applications	Invoke Set Allowed Application • ProfileID set to 0 • AppIDs is set to “*”	• No error response

2.3 Client Definitions

2.3.1 UPnP Control Point Connect

This definition contains all necessary steps to make a UPnP “Connection” to the UPnP Control Point.

Step	Name	Description	Expected Result
1	UPnP Connect	SSDP:alive advertisements Note: Send UPnP Bye-Bye message, prior UPnP connect, if UPnP Server is still operational.	• Control Point checks for Device XML • Control Point MAY use M-Search instead
2	UPnP Device Description	Provide Server Device XML	• Retrieve Service XML

Step	Name	Description	Expected Result
3	UPnP Service Description	Provide Server Service XML	

3 SERVER FEATURE TEST CASES

3.1 Notification Service

3.1.1 SR/UPNP/DEVICE/NotificationService

Requirement: CONDITIONAL

Condition: Server support UPnP Notification Server Service

Test if Server supports Notification Server Service in the Device XML.

Step	Name	Description	Expected Result
1	UPnP Connect	Preparing the UPnP connection by making an initialization, registering the client and waiting for the device to announce itself.	
2	UPnP MSearch	Send MSearch request	• Device announces itself
3	UPnP Device Description	Test the service description for parsable XML formating and availability of service types and their control and event URLs.	• Valid device description (according to specification) • Support for TmNotificationServer:1 service

Table 1: Notification Service Support

3.2 Notification Server Actions

3.2.1 SR/UPnP/APP/GetSupportedApplication

Requirement: CONDITIONAL

Condition: Server support UPnP Notification Server Service

Test if the server provides a list of applications, supporting notification. The Test Engineer MAY need to launch applications on the phone.

Step	Name	Description	Expected Result
1	UPnP Server Connect	See definitions in [3]	
2	UPnP Application Launch	Test Engineer MAY need to launch applications supporting notifications. • Invoke Get Application List action • Invoke Launch application action for application(s) to be started. This step MAY be skipped.	• Provide application list (if needed) • Launch applications (if needed)
3	UPnP Get Supported Applications	Invoke Get Supported Application • ProfileID set to 0	• Receive comma separated list of application ids. • Must be unique • May be empty
4	UPnP Server Disconnect	See definitions in [3]	

Table 2: Provide List of Applications supporting Notification

3.2.2 SR/UPnP/APP/SetAllowedApplication

Requirement: CONDITIONAL

Condition: Server support UPnP Notification Server Service

Test if the server registers notifications for a list of applications, supporting notification. The Test Engineer MAY need to launch applications on the phone.

Step	Name	Description	Expected Result
1	UPnP Server Connect	See definitions in [3]	
2	UPnP Application Launch	Test Engineer MAY need to launch applications supporting notifications. • Invoke Get Application List action • Invoke Launch application action for application(s) to be started. This step MAY be skipped.	• Provide application list (if needed) • Launch applications (if needed)
3	UPnP Get Supported Applications	Invoke Get Supported Application • ProfileID set to 0	• Receive comma separated list of application ids. • Must be unique • May be empty
4	UPnP Set Allowed Applications	Invoke Set Allowed Application • ProfileID set to 0 • AppIDs is set to “*”	• No error response
5	UPnP Server Disconnect	See definitions in [3]	

Table 3: Set List of Applications, providing Notifications

3.2.3 SR/UPnP/APP/GetNotification

Requirement: CONDITIONAL

Condition: Server support UPnP Notification Server Service AND

Server has installed applications, supporting Notification

Test if the server provides information on notifications. The test engineer MAY be required to perform known tasks to trigger a notification.

Step	Name	Description	Expected Result
1	UPnP Server Register Notifications	See definitions	
2	Subscribe to ActiveNotiEvent event	Subscribe to the App Status Update event	• Receive initial event
3	Notification	Wait for incoming notification event Invoke GetNotification action for active notification	• Active Noti Event received • Receive valid Notification XML
4	UPnP Server Disconnect	See definitions in [3]	

Table 4: Get Notification Information

3.2.4 SR/UPnP/APP/LaunchVncOnNotification

Requirement: CONDITIONAL

Condition: Server support UPnP Notification Server Service AND

Server has installed applications, supporting Notification

Test if the server can launch the application the notification originated from. The test engineer MAY be required to perform known tasks to trigger a notification.

Step	Name	Description	Expected Result
1	UPnP Server Register Notifications	See definitions	
2	Subscribe to ActiveNotiEvent event	Subscribe to the App Status Update event	• Receive initial event
3	Notification	Wait for incoming notification event Invoke GetNotification action for active notification	• Active Noti Event received • Receive valid Notification XML
4	Launch VNC	Invoke Launch Application with the AppId provided in Notification XML. Test engineer MUST be able to see the notification on the DUT.	• Receive URL to VNC server
5	Terminate VNC	Terminate Application	
6	UPnP Server Disconnect	See definitions in [3]	

Table 5: Launch VNC on Notification

3.2.5 SR/UPnP/APP/InvokeVncAction

Requirement: CONDITIONAL

Condition: Server support UPnP Notification Server Service

Server has installed applications, supporting Notification

Test if the server does show the notification on its screen, when the application, which is originating the notification is launched. The test engineer MAY be required to perform known tasks to trigger a notification.

Step	Name	Description	Expected Result
1	UPnP Server Register Notifications	See definitions	
2	Subscribe to Active-NotiEvent event	Subscribe to the App Status Update event	• Receive initial event
3	Notification	Wait for incoming notification event Invoke GetNotification action for active notification	• Active Noti Event received • Receive valid Notification XML
4	Launch VNC	Invoke Launch Application with the AppId provided in Notification XML. Test engineer MUST be able to see the notification on the DUT.	• Receive URL to VNC server
5	VNC Server Handshake	See Definitions in [4]	
6	VNC Server Configuration	See Definitions in [4]	
7	VNC Server Start Operation	See Definitions in [4]	
8	VNC base UI	Test engineer confirms that Notification UI is visible on the DUT	• Notification UI visible on Client Display. • Same as UI on DUT (test engineer to confirm) • Receive Active-NotiEvent event (either empty string or new NotiID); Note: event can happen in steps 5-8.
9	Terminate VNC	Terminate Application	
10	UPnP Server Disconnect	See definitions in [3]	

Table 6: Invoke VNC on Received Notification

3.2.6 SR/UPnP/APP/InvokeNotiAction

Requirement: CONDITIONAL

Condition: Server support UPnP Notification Server Service AND

Server has installed applications, supporting Notification AND

Notification supporting Client Action UI

Test if the server can handle actions on the notification provided. The test engineer MAY be required to perform known tasks to trigger a notification. Test engineer will need to know, how the DUT reacts on the different notification actions.

Step	Name	Description	Expected Result
1	UPnP Server Register Notifications	See definitions	
2	Subscribe to Active-NotiEvent event	Subscribe to the App Status Update event	• Receive initial event
3	Notification	Wait for incoming notification event Invoke GetNotification action for active notification	• Active Noti Event received • Receive valid Notification XML
4	Invokie Noti Action	Invoke Noti Action for one of the provided actions (random pick). Launch the application (if required).	• DUT handles the notification. • Receive URL (if required) • Receive Active-NotiEvent event (either empty string or new NotiID).
5	UPnP Server Disconnect	See definitions in [3]	

Table 7: Invoke Action on Received Notification

3.3 Notification Server Events

3.3.1 SR/UPNP/APP/ActiveNotiEventSubscribe

Requirement: CONDITIONAL

Condition: Server support UPnP Notification Server Service

Test if the control point can subscribe to the Active Noti Event event and if the server provides the valid initial response.

Step	Name	Description	Expected Result
1	UPnP Server Connect	See definitions in [3]	
2	Subscribe to Active Noti Event event	Subscribe to the Active Noti Event event	• Receive initial event
3	UPnP Server Disconnect	See definitions in [3]	

Table 8: Active Noti Event – Subscribe

3.3.2 SR/UPNP/APP/ActiveNotiEventEvent

Requirement: CONDITIONAL

Condition: Server support UPnP Notification Server Service AND

Server has installed applications, supporting Notification

Test the Active Noti Event event. Test Engineer is asked to execute the known steps to trigger and event.

Step	Name	Description	Expected Result
1	UPnP Server Connect	See definitions in [3]	
2	Subscribe to Active Noti Event event	Subscribe to the App Status Update event	• Receive initial event
3	Check Acitive Noti Event	Test Engineer executes known steps to trigger a notification.	• Receive update event
4	UPnP Server Disconnect	See definitions in [3]	

Table 9: Active Noti Event – Event

3.3.3 SR/UPNP/APP/NotiAppListUpdateSubscribe

Requirement: CONDITIONAL

Condition: Server support UPnP Notification Server Service

Test if the control point can subscribe to the NotiAppListUpdate event and if the server provides the valid initial response.

Step	Name	Description	Expected Result
1	UPnP Server Connect	See definitions in [3]	
2	Check Noti App List Update event	Test Engineer executes known steps to start an application, which will provide new notifications	• Receive initial event

Step	Name	Description	Expected Result
3	Subscribe to Noti App List Update event	Subscribe to the NotiAppListUpdate event	

Table 10: Noti App List Update – Subscribe

3.3.4 SR/UPNP/APP/NotiAppListUpdateEvent

Requirement: CONDITIONAL

Condition: Server support UPnP Notification Server Service AND
Server has installed applications, supporting Notification

Test the NotiAppListUpdate event. Test Engineer is asked to execute the known steps to trigger and event.

Step	Name	Description	Expected Result
1	UPnP Server Connect	See definitions in [3]	
2	Subscribe to Noti App List Update event	Subscribe to the NotiAppListUpdate event	• Receive initial event
3	Check Noti App List Update event	Test Engineer executes known steps to start an application, which will provide new notifications	• Receive update event
4	UPnP Server Disconnect	See definitions in [3]	

Table 11: Noti App List Update – Event

3.4 PICS Validation

The PICS validation test cases will independently detect the existence of MirrorLink features in the DUT. All features, which are detectable, could in practice be used from a connected MirrorLink device, and are therefore subject to validation in the certification program through other test cases. Hence the objective of the PICS validation test cases is not to assess whether the feature is implemented correctly, but to collect supported features from the DUT and to check this against the entries made in the PICS document.

A feature, which is detected, but marked as "`not implemented`" in the PICS document will fail the test case. A feature, which is not detected, but marked as "`implemented`" in the PICS document, will fail the test case.

3.4.1 SR/UPNP/NOTI/PICS/ServiceXml

Requirement: MANDATORY

Condition: None

This test case validates the PICS entries with respect to the Notification Server Service XML settings.

Step	Name	Description	Expected Result
1	UPnP Server Connect	See Definitions	
2	Check PICS feature	FEAT_SERVER_UPNP_Notification_State_NotEvent	• `ActiveNotiEvent` listed as evented variable in Notification Server Service description

Step	Name	Description	Expected Result
3	Check PICS feature	FEAT_SERVER_UPNP_Notification_State_NotiAppListEvent	• NotiAppListUpdate listed as evented variable in Notification Server Service description
4	Check PICS feature	FEAT_SERVER_UPNP_Notification_Service_Get	• GetNotification listed as action in Notification Server Service description.
5	Check PICS feature	FEAT_SERVER_UPNP_Notification_Service_GetApps	• GetSupportedApplications listed as action in Notification Server Service description.
6	Check PICS feature	FEAT_SERVER_UPNP_Notification_Service_SetApp	• SetAllowedApplications listed as action in Notification Server Service description.
7	Check PICS feature	FEAT_SERVER_UPNP_Notification_Service_Invoke	• InvokeNotiAction listed as action in Notification Server Service description.

Table 12: MirrorLink Server Notification Server Service XML settings PICS Checkup

4 CLIENT FEATURE TEST CASES

4.1 Notification Server Services

4.1.1 CL/UPNP/APP/GetSupportedApplication

Requirement: CONDITIONAL

Condition: Client support UPnP Notification Server Service

Tests if the UPnP Control Point is requesting application listing.

Step	Name	Description	Expected Result
1	UPnP Control Point Connect	See definitions	
2	UPnP Action Get Supported Application	Wait for Get Supported Application action. Provide list of at least 3 applications.	• Valid Get Supported Application action received

Table 13: Get List of Applications, supporting Notifications

4.1.2 CL/UPNP/APP/SetAllowedApplication

Requirement: CONDITIONAL

Condition: Client support UPnP Notification Server Service

Tests if the UPnP Control Point is setting notification list.

Step	Name	Description	Expected Result
1	UPnP Control Point Connect	See definitions	
2	UPnP Action Get Supported Application	Wait for Get Supported Application action. Provide list of at least 3 applications. Test Engineer MAY need to know, the application ids, from which the DUT will accept notifications.	• Valid Get Supported Application action received
3	UPnP Action Set Allowed Application	Wait for Set Allowed Application.	• Valid Set Allowed Application action received • At least 1 application set

Table 14: Set List of Application, providing Notifications

4.1.3 CL/UPNP/APP/GetNotification

Requirement: CONDITIONAL

Condition: Client support UPnP Notification Server Service AND
Client supports GetNotification

Tests if the UPnP Control Point is receiving a notification.

Step	Name	Description	Expected Result
1	UPnP Control Point Connect	See definitions	
2	UPnP Action Get Application List	Wait for Get Supported Application action. Provide list of at least 3 applications.	• Valid Get Supported Application action received

Step	Name	Description	Expected Result
		Test Engineer MAY need to know, the application ids, from which the DUT will accept notifications.	
3	UPnP Action Set Allowed Application	Wait for Set Allowed Application.	• Valid Set Allowed Application action received • At least 1 application set
4	Get Notification	Trigger notification event (Client MUST have subscribed to the notification event prior to this)	• Valid Get Notification action received • Valid Notification ID

Table 15: Trigger Notification Event

4.1.4 CL/UPNP/APP/GetNotificationWithVncAction

Requirement: CONDITIONAL

Condition: Client support UPnP Notification Server Service

Test if the UPnP Control Point receives a notification and launches the application.

Step	Name	Description	Expected Result
1	UPnP Control Point Connect	See definitions	
2	UPnP Action Get Application List	Wait for Get Supported Application action. Provide list of at least 3 applications. Test Engineer MAY need to know, the application ids, from which the DUT will accept notifications.	• Valid Get Supported Application action received
3	UPnP Action Set Allowed Application	Wait for Set Allowed Application.	• Valid Set Allowed Application action received • At least 1 application set
4	Get Notification	Trigger notification event (Client MUST have subscribed to the notification event prior to this)	DUT MAY request notification details: • Valid Get Notification action received • Valid Notification ID
5	Launch Application	Wait for client to launch application. DUT MAY clear the notification, without launching the application	• Valid Launch Application • Application ID matches OR • `InvokeNotiAction` received with zero `actionID` (0x0000) • `NotiID` matches

Table 16: Launch Application on Notification

4.1.5 CL/UPNP/APP/GetNotificationWithInvokeAction

Requirement: CONDITIONAL

Condition: Client support UPnP Notification Server Service AND

Client supports Notification UI

Test if the UPnP Control Point is receiving a notification and launching the application.

Step	Name	Description	Expected Result
1	UPnP Control Point Connect	See definitions	
2	UPnP Action Get Supported Application	Wait for Get Supported Application action. Provide list of at least 3 applications. Test Engineer MAY need to know, the application ids, from which the DUT will accept notifications.	• Valid Get Supported Application action received
3	UPnP Action Set Allowed Application	Wait for Set Allowed Application.	• Valid Set Allowed Application action received • At least 1 application set
4	Get Notification	Trigger notification event (Client MUST have subscribed to the notification event prior to this)	• Valid Get Notification action received • Valid Notification ID
5	Launch Application	Wait for InvokeNotiAction	• Valid Invoke NotiAction Received • Notification ID matches • DUT shows Notification UI

Table 17: Invoke Notification Action on Notification

4.1.6 CL/UPNP/APP/NotificationSignatureSuccess

Requirement: CONDITIONAL

Condition: Client support UPnP Notification Server Service AND
Client supports GetNotification AND
Client validates XML signature

Test if the UPnP Control Point validates the XML signature.

Step	Name	Description	Expected Result
1	UPnP Control Point Connect	See definitions	
2	DAP Client Launch	See Definitions in [5]	
3	DAP Attestation Request	Device attestation request	• Receive Device Attestation Request • Includes UPnP Server or “*” • Use CCC trust root
4	DAP Attestation Response	Provide UPnP application public key in Device Attestation Response	• Terminate DAP server (optional) • Retrieve application listing
5	UPnP Action Get Application List	Wait for Get Supported Application action. Provide list of at least 3 applications. Test Engineer MAY need to know, the application ids, from which the DUT will accept notifications.	• Valid Get Supported Application action received
6	UPnP Action Set Allowed Application	Wait for Set Allowed Application.	• Valid Set Allowed Application action received • At least 1 application set

Step	Name	Description	Expected Result
7	Get Notification	Trigger notification event (Client MUST have subscribed to the notification event prior to this)	• Valid Get Notification action received • Valid Notification ID
8	Validation Notification	Provide notification with correct XML signature	• Notification is handled correctly

Table 18: Validation of Notification XML Signature – Success

4.1.7 CL/UPNP/APP/NotificationSignatureFailure

Requirement: CONDITIONAL

Condition: Client support UPnP Notification Server Service AND
Client supports GetNotification AND
Client validates XML signature

Test if the UPnP Control Point validates the XML signature.

Step	Name	Description	Expected Result
1	UPnP Control Point Connect	See definitions	
2	DAP Client Launch	See Definitions in [5]	
3	DAP Attestation Request	Device attestation request	• Receive Device Attestation Request • Includes UPnP Server or "*" • Use CCC trust root
4	DAP Attestation Response	Provide UPnP application public key in Device Attestation Response	• Terminate DAP server (optional) • Retrieve application listing
5	UPnP Action Get Application List	Wait for Get Supported Application action. Provide list of at least 3 applications. Test Engineer MAY need to know, the application ids, from which the DUT will accept notifications.	• Valid Get Supported Application action received
6	UPnP Action Set Allowed Application	Wait for Set Allowed Application.	• Valid Set Allowed Application action received • At least 1 application set
7	Get Notification	Trigger notification event (Client MUST have subscribed to the notification event prior to this)	• Valid Get Notification action received • Valid Notification ID
8	Validation Notification	Provide notification with incorrect XML signature	• Notification is ignored • InvokeNotiAction with ActionID=0x00 received

Table 19: Validation of Notification XML Signature – Failure

4.2 Notification Server Events

4.2.1 CL/UPNP/APP/ActiveNotiEventSubscribe

Requirement: CONDITIONAL

Condition: Client support UPnP Notification Server Service

Test if the UPnP Control Point subscribes to the evented variable.

Step	Name	Description	Expected Result
1	UPnP Control Point Connect	See definitions	
2	Subscribe to Active Noti Event event	User is asked to execute known steps to start eventing	• Receive event subscription for Active Noti Event

Table 20: Subscribe to Notification

4.2.2 CL/UPNP/APP/NotiAppListUpdateSubscribe

Requirement: CONDITIONAL

Condition: Client support UPnP Notification Server Service

Test if the UPnP Control Point subscribes to the evented variable.

Step	Name	Description	Expected Result
1	UPnP Control Point Connect	See definitions	
2	Subscribe to Noti App List Update event	User is asked to execute known steps to start eventing	• Receive event subscription for Noti App List Update Event

Table 21: Subscribe to Notification

4.2.3 CL/UPNP/APP/NotiAppListUpdateEvent

Requirement: CONDITIONAL

Condition: Client support UPnP Notification Server Service AND

Client checks Supported Applications on received NotiAppListUpdate event

Test if the UPnP Control Point subscribes to the evented variable.

Step	Name	Description	Expected Result
1	UPnP Control Point Connect	See definitions	
2	Subscribe to Noti App List Update event	User is asked to execute known steps to start eventing	• Receive event subscription for Noti App List Update Event
3	UPnP Action Get Supported Application	Send NotiAppListUpdateEvent adding at least two new applications and remove one application.	• Valid Get Supported Application action received

Table 22: Act on Notification

5 REFERENCES

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

[2] Car Connectivity Consortium, “MirrorLink – UPnP Notification Server Service”, Version 1.1; CCC-TS-028

[3] Car Connectivity Consortium, “MirrorLink – UPnP Server Device Test Specification”, Version 1.1; CCC-TS-031

[4] Car Connectivity Consortium, “MirrorLink – VNC based Display and Control Test Specification”, Version 1.1; CCC-TS-011

[5] Car Connectivity Consortium, “MirrorLink – Device Attestation Test Specification”, Version 1.1, CCC-TS-015