Car Connectivity Consortium

MirrorLink®

UPnP Application Server Service Test Specification

Version 1.1.9
(CCC-TS-025)

VERSION HISTORY

Version	Date	Comment
1.1	31 March 2012	Approved Version
1.1.1	16 October 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	17 June 2014	Approved Errata Version
1.1.5	16 October 2014	Approved Errata Version
1.1.6	10 November 2014	Approved Errata Version
1.1.7	18 March 2015	Approved Errata Version
1.1.8	17 June 2015	Approved Errata Version
1.1.9	27 August 2015	Approved Errata Version

LIST OF CONTRIBUTORS

Brakensiek, Jörg (Editor)	Car Connectivity Consortium LLC
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 Application 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

None.

2.3 Client Definitions

None.

3 SERVER FEATURE TEST CASES

3.1 Application Server Actions

3.1.1 SR/UPnP/APP/GetApplicationList

Requirement: MANDATORY

Condition: None

Invoking the 'GetApplicationList' action on the TmApplicationServer:1 service and checking for validity of the XML.

Step	Name	Description	Expected Result
1	UPnP Server Connect	See definitions in [4]	
2	UPnP Action GetApplication-List	Invoking Get Application List action • Empty AppListingFilter • ProfileID set to 0	• Receive valid XML, • Includes all optional element
3	UPnP Action GetCertifiedAppListing	Invoke GetCertifiedAppListing • Empty AppCertFilter • ProfileID set to 0	• Receive application list
4	Check App Certificate Values	CTS downloads the application certificates from the provided `appCertificateURL`.	• Certificate of base or drive certified apps can be downloaded • Entries from the app certificate match the entries in the app listing • Trust level is set to 0xA0 in case the app is certified (test step 3). Note: This checking does not apply to Developer Certificates.
5	UPnP Server Disconnect	See definitions in [4]	

Table 1: UPnP Action GetApplicationList – Test Steps

3.1.2 SR/UPNP/APP/AppListingFilter

Requirement: MANDATORY

Condition: None

Test the functionality of the AppListeningFilter. This test requires at least one application being advertised, with a valid provider name. In case no other application exists, install the Common API test application.

Step	Name	Description	Expected Result
1	UPnP Server Connect	See definitions in [4]	
2	UPnP Action GetApplication-List	Invoke Get Application List action • Empty AppListingFilter • ProfileID set to 0	• Receive valid XML
3	UPnP App Listing Filter Wildcard	Invoke Get Application List action • AppListingFilter set to name=”*” • ProfileID set to 0	• Receive valid XML

Step	Name	Description	Expected Result
			• Contains all applications from previous result
4	UPnP App Listing Filter Single match	Invoke Get Application List action • AppListingFilter set to include name with an entry available from the list in step 2. • ProfileID set to 0	• Receive valid XML, formatted according to the specification • Contains all matching applications
5	UPnP App Listing Filter Dual match	Invoke Get Application List action • AppListingFilter set to include name and providerName with entries available from the list in step 2. • ProfileID set to 0	• Receive valid XML • Contains all matching application
6	UPnP Server Disconnect	See definitions in [4]	

Table 2: UPnP Action AppListingFilter – Test Steps

3.1.3 SR/UPNP/APP/AppListingFilterRemoteProtocolID

Requirement: MANDATORY

Condition: None

Test the functionality of the `AppListeningFilter`. This test case validates, whether the MirrorLink Server correctly responds to search requests, following the identification guidelines in the UPnP Application Server service specification to identify VNC apps, RTP Servers and Clients, and CDB and DAP endpoints.

Step	Name	Description	Expected Result
1	UPnP Server Connect	See definitions in [4]	
2	UPnP Action GetApplicationList	Invoke Get Application List action • Empty `AppListingFilter` • `ProfileID` set to 0	• Receive valid XML
3	UPnP App Listing Filter VNC apps	Invoke Get Application List action • Set `AppListingFilter` to ‟`protocolId="VNC"`”. • `ProfileID` set to 0	• Receive valid XML, formatted according to the specification • Contains all VNC applications
4	UPnP App Listing Filter RTP Servers	Invoke Get Application List action • Set `AppListingFilter` to ‟`protocolId="RTP", appCategory="0xF0000001"`”. • `ProfileID` set to 0	• Receive valid XML, formatted according to the specification • Contains all RTP Servers
5	UPnP App Listing Filter RTP Clients	Invoke Get Application List action • Set `AppListingFilter` to ‟`protocolId="RTP", appCategory="0xF0000002"`”. • `ProfileID` set to 0	• Receive valid XML, formatted according to the specification • Contains all RTP Clients • MUST be empty, if RTP Client not supported.
6	UPnP App Listing Filter DAP Endpoing	Invoke Get Application List action • Set `AppListingFilter` to	• Receive valid XML, formatted according to the specification

Step	Name	Description	Expected Result
		"protocolId="DAP", appCategory="0xF0000001"". • `ProfileID` set to 0	• Contains DAP end-point(s)
7	UPnP App Listing Filter CDB Endpoint	Invoke Get Application List action • Set `AppListingFilter` to "protocolId="CDB", appCategory="0xF0000000"". • `ProfileID` set to 0	• Receive valid XML, formatted according to the specification • Contains all CDB end-point(s) • MUST be empty, if CDB not supported.
8	UPnP Server Disconnect	See definitions in [4]	

Table 3: UPnP Action AppListingFilter – Remote Protocol ID

3.1.4 SR/UPNP/APP/LaunchApplication

Requirement: MANDATORY

Condition: None

Invoking the Launch Application action for every VNC based application within the application list and checking for validity on - the action launch - the return code.

Step	Name	Description	Expected Result
1	UPnP Server Connect	See definitions in [4]	
2	UPnP Get Application List	Invoking Get Application List action • Empty AppListingFilter • ProfileID set to 0	• Receive valid XML, formatted according to the specification,
3	Subscribe to UPnP event	Subscribe to the Application Server service's events	• Receive initial UPnP `AppListUpdate` event • Receive initial UPnP `AppStatusUpdate` event.
4	UPnP Launch Application	Invoking Launch Application action for VNC UI based application within the application list. The launched application must not already be in the foreground. Note: Individual applications may not bring a UI into the foreground (PIXIT).	• Application is started on the MirrorLink Server • Return URL • On VNC connection, the DUT does not replicates the UI of the previous foreground application.
5	Check for UPnP events	Check for Application Server service events.	• No `AppListUpdate` event received (Note – if an `AppListUpdate` is received, CTS MUST check that the `appList` actually changed) • `AppStatusUpdate` event received • Note: May receive more than one `AppStatusUpdate` event. Event MAY contain another application.

Step	Name	Description	Expected Result
6	UPnP Get Application Status	Check Application Status of launched app.	• Launched application is in `foreground`.
7	Repeat	Go to step 4 for next VNC based application	
8	UPnP Server Disconnect	See definitions in [4]	

Table 4: UPnP Action LaunchApplication – Test Steps

3.1.5 SR/UPNP/APP/TerminateApplication

Requirement: MANDATORY

Condition: None

Test whether any VNC based application can be terminated via Terminate Application action, after being previously launched.

Step	Name	Description	Expected Result
1	UPnP Server Connect	See definitions in [4]	
2	UPnP Get Application List	Invoking Get Application List action • Empty AppListingFilter • ProfileID set to 0	• Receive valid XML, formatted according to the specification,
3	Subscribe to Update events	Subscribe to the Application Server service's events	• Receive initial events
4	UPnP Launch Application	Invoke Launch Application action for VNC UI based application within the application list. Note: Individual applications may not bring a UI into the foreground (PIXIT).	• Application is started on the MirrorLink Server • Return URL
5	UPnP Terminate Application	Invoke Terminate Application action for the previously launched application	• Application is not in the foreground on the MirrorLink Server
6	Check for UPnP events	Check for Application Server service events. Note: Individual applications may not terminate and stay in the foreground (PIXIT).	• No `AppListUpdate` event received (Note – if an `AppListUpdate` is received, CTS MUST check that the `appList` actually changed) • `AppStatusUpdate` event received, if app can be terminated. • Note: May receive more than one `AppStatusUpdate` event. Event MAY contain another application.
7	UPnP Get Application Status	Check Application Status of terminated app.	• Terminated application is in `background` or `notRunning`, if app can be terminated.

Step	Name	Description	Expected Result
8	Repeat	Go back to step 3 for next VNC based application	
9	UPnP Server Disconnect	See definitions in [4]	

Table 5: UPnP Action Terminate Application – Test Steps

3.1.6 SR/UPNP/APP/TerminateAudioApplication

Requirement: MANDATORY

Condition: None

Test whether any VNC based audio application can be terminated via Terminate Application action, after being previously launched. The termination must also terminate the audio stream. The test will only handle applications, which provide an audio stream.

Step	Name	Description	Expected Result
1	UPnP Server Connect	See definitions in [4]	
2	UPnP Get Application List	Invoking Get Application List action • Empty AppListingFilter • ProfileID set to 0	• Receive valid XML, formatted according to the specification,
3	UPnP Launch Application	Invoke Launch Application action for VNC based audio application within the application list.	• Application is started on the MirrorLink Server • Return URL
4	Start Audio	Test engineer executes the known steps to start the audio playback.	• Audio stream received
5	UPnP Terminate Application	Invoke Terminate Application action for the previously launched application Go back to step 3 for next VNC based audio application	• Application is not in the foreground on the MirrorLink Server • Audio stream is stopped • Audio is not playing via the DUT
6	UPnP Server Disconnect	See definitions in [4]	

Table 6: UPnP Action Terminate Audio Application

3.1.7 SR/UPNP/APP/GetApplicationStatus

Requirement: MANDATORY

Condition: None

Invoking the Get Application Status action and checking for validity on - the returned xml itself - every 'appStatus' element has - a valid 'appID' element - a 'status' element - every 'status' element has - a 'profileID' element - a 'statusType' element with value from 'Foreground', 'Background', 'Notrunning'

Step	Name	Description	Expected Result
1	UPnP Server Connect	See definitions in [4]	
2	Subscribe to App Status Update event	Subscribe to the App Status Update event	• Receive initial event

Step	Name	Description	Expected Result
3	UPnP Get Application List	Invoking Get Application List action - Empty AppListingFilter - ProfileID set to 0	- Receive valid XML, formatted according to the specification,
4	UPnP Launch Application	Invoke Launch Application action for VNC UI based application within the application list.	- Application is started on the MirrorLink Server - Return URL
5	UPnP App Status Update event	Wait for UPnP App Status Update event to be received.	- No `AppListUpdate` event received (Note – if an `AppListUpdate` is received, CTS MUST check that the `appList` actually changed) - `AppStatusUpdate` event received. - Note: May receive more than one `AppStatusUpdate` event. - Events include application from step 4.
6	UPnP Get Application Status	Invoke Get Application Status action for the previously launched application	- First Get Application Status response is “`Foreground`”
7	UPnP Terminate Application	Invoke Terminate Application action for the previously launched application (step 5) Note: Individual applications may not terminate and stay in the foreground (PIXIT).	
8	UPnP Get Application Status	Invoke Get Application Status action for the previously terminated application (step 7)	- Second Get Application Status response is “`Notrunning`” or “`Background`”, if app can be terminated
9	Repeat	Go to step 4 for next VNC UI based application.	
10	UPnP Get Application Status	Invoke Get Application Status action for all applications using “`*`” as the application identifier.	- Receive Get Application Status response - Response includes application status of all advertised apps.
11	UPnP Server Disconnect	See definitions in [4]	

Table 7: UPnP Action GetApplicationStatus – Test Steps

3.2 Application Server Events

3.2.1 SR/UPNP/APP/AppStatusUpdateSubscribe

Requirement: MANDATORY

Condition: None

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

Step	Name	Description	Expected Result
1	UPnP Server Connect	See definitions in [4]	
2	UPnP Get Application List	Invoking Get Application List action • Empty AppListingFilter • ProfileID set to 0	• Receive valid XML, formatted according to the specification
3	Subscribe to App Status Update event	Subscribe to the App Status Update event	• Receive initial event
4	UPnP Server Disconnect	See definitions in [4]	

Table 8: UPnP Event AppStatusUpdate Subscribe – Test Steps

3.2.2 SR/UPNP/APP/AppStatusUpdateEvent

Requirement: MANDATORY

Condition: None

Test the App Status Update event by launching or terminating any VNC based application and wait for an incoming event.

Randomly pick three UI-based applications A, B and C. Note: Individual applications may not change its application status (PIXIT). Those application must not be used. Include the Home Screen application, if available.

Step	Name	Description	Expected Result
1	UPnP Server Connect	See definitions in [4]	
2	UPnP Get Application List	Invoking Get Application List action • Empty AppListingFilter • ProfileID set to 0	• Receive valid XML, formatted according to the specification
3	Subscribe to Update event	Subscribe to the Application Server service's events	• Receive initial event • `AppStatusUpdate` event lists all advertised apps.
4	UPnP Get Application Status	Invoke Get Application Status action for all listed VNC applications.	• All: `notRunning` or `background` • DUT foreground app (if advertised): `foreground`
5	UPnP Launch application	Invoke Launch Application action for app A	• No `AppListUpdate` event received (Note – if an `AppListUpdate` is received, CTS MUST check that the `appList` actually changed)

Step	Name	Description	Expected Result
			• AppStatusUpdate event received. • Note: May receive more than one AppStatusUpdate event. • Event includes app A. • Event may include one other app (step 4).
6	UPnP Get Application Status	Invoke Get Application Status action for all listed applications.	• App A: foreground. • Other: background
7	UPnP Launch application	Invoke Launch Application action for app B	• No AppListUpdate event received (Note – if an AppListUpdate is received, CTS MUST check that the appList actually changed) • AppStatusUpdate event received. • Note: May receive more than one AppStatusUpdate event. • Event includes only app A and B.
8	UPnP Get Application Status	Invoke Get Application Status action for all listed applications.	• App A: background. • App B: foreground
9	UPnP Launch application	Invoke Launch Application action for app C	• No AppListUpdate event received (Note – if an AppListUpdate is received, CTS MUST check that the appList actually changed) • AppStatusUpdate event received. • Note: May receive more than one AppStatusUpdate event. • Event includes only app B and C.
10	UPnP Get Application Status	Invoke Get Application Status action for all listed applications.	• App B: background. • App C: foreground
11	UPnP Terminate application	Invoke Terminate Application action for app A Note: Individual applications may not terminate and stay in the foreground (PIXIT).	• No AppListUpdate event received (Note – if an AppListUpdate is received, CTS MUST check that the appList actually changed) • AppStatusUpdate event received, if app can be terminated. • Event includes only app A.
12	UPnP Get Application Status	Invoke Get Application Status action for all listed applications.	• App A: notRunning or background
13	UPnP Terminate application	Invoke Terminate Application action for app C	• No AppListUpdate event received (Note – if an AppListUpdate is received,

Step	Name	Description	Expected Result
		Note: Individual applications may not terminate and stay in the foreground (PIXIT).	CTS MUST check that the `appList` actually changed) • `AppStatusUpdate` event received, if app can be terminated • Note: May receive more than one `AppStatusUpdate` event. • Event includes app C. • Event may include one other app. • Other App either app B or Home Screen app.
14	UPnP Get Application Status	Invoke Get Application Status action for all listed applications.	• App C: `notRunning` or `background` • Other: `foreground`
15	UPnP Server Disconnect	See definitions in [4]	

Table 9: UPnP Event AppStatusUpdate – Test Steps

3.2.3 SR/UPNP/APP/AppListUpdateSubscribe

Requirement: MANDATORY

Condition: None

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

Step	Name	Description	Expected Result
1	UPnP Server Connect	See definitions in [4]	
2	UPnP Get Application List	Invoking Get Application List action • Empty AppListingFilter • ProfileID set to 0	• Receive valid XML, formatted according to the specification,
3	Subscribe to App List Update event	Subscribe to the App List Update event	• Receive initial event
4	UPnP Server Disconnect	See definitions in [4]	

Table 10: UPnP Event AppListUpdate Subscribe – Test Steps

3.2.4 SR/UPNP/APP/AppListUpdateEvent

Requirement: CONDITIONAL

Condition: Server supports dynamic update of Application List

Test the App List Update event by executing known steps on the MirrorLink server.

Step	Name	Description	Expected Result
1	UPnP Server Connect	See definitions in [4]	
2	UPnP Get Application List	Invoking Get Application List action • Empty AppListingFilter • ProfileID set to 0	• Receive valid XML, formatted according to the specification,

Step	Name	Description	Expected Result
3	Subscribe to App Status Update event	Subscribe to the App List Update event	• Receive initial event
4	Check App List Update	Conduct the necessary steps on the Server to create a new application list.	• Receive update event
5	Check Application List	Invoke Get Application List action • Empty AppListingFilter • ProfileID set to 0	• Receive valid XML, formatted according to the specification • XML is different from first XML
6	UPnP Server Disconnect	See definitions in [4]	

1 Table 11: UPnP Event AppListUpdate– Test Steps

3.3 Content Attestation

3.3.1 SR/UPNP/APP/ContentAttestation

Requirement: CONDITIONAL

Condition: Server supports UPnP Server Attestation AND

Server supports XML signature

Requests the Application List via UPnP and uses DAP for Attestation with the public key

Step	Name	Description	Expected Result
1	UPnP Server Connect	See definitions in [4]	
2	DAP Initialize	See DAP definitions [3]	
3	DAP UPnP Server attestation	Attestation of the TerminalMode:UPnP-Server	• Valid response • Receive UPnP session key
4	DAP disconnect	See DAP definitions [3]	
5	UPnP Get Application List	Invoking Get Application List action • Empty AppListingFilter • ProfileID set to 0	• Receive valid XML, formatted according to the specification • XML signed • XML signature verified with UPnP session key
6	UPnP Server Disconnect	See definitions in [4]	

Table 12: UPnP Content Attestation – Test Steps

3.4 Certification

3.4.1 SR/UPNP/CERT/GetCertifiedApplicationsList

Requirement: MANDATORY

Condition: None

Invoking the 'GetCertifiedApplicationList' action on the TmApplicationServer:1 service and checking for validity of the XML.

Step	Name	Description	Expected Result
1	UPnP Server Connect	See definitions in [4]	
2	UPnP Action GetApplication-List	Invoke Get Applications List action • Empty AppListingFilter • ProfileID set to 0	• Receive valid XML
3	UPnP Action GetCertifiedApplicationList	Invoke Get Certified Applications List action • Empty CertAppListingFilter • ProfileID set to 0	• Receive comma-separated list of appIDs • All applications on the certified applications list are included in the Get Application List (as received in Step 2)
4	UPnP Server Disconnect	See definitions in [4]	

Table 13: UPnP Action GetCertifiedApplicationsList – Test Steps

3.4.2 SR/UPNP/CERT/AppCertFilter

Requirement: MANDATORY

Condition: None

Test the functionality of the AppCertFilter.

Step	Name	Description	Expected Result
1	UPnP Server Connect	See definitions in [4]	
2	UPnP Action GetCertifiedApplicationList	Invoke Get Certified Applications List action • Empty AppCertFilter • ProfileID set to 0	• Receive valid XML, formatted according to the specification
3	UPnP Cert App Listing Filter Wildcard	Invoke Get Certified Applications List action • AppCertFilter set to name=”*” • ProfileID set to 0	• Receive comma-separated list of appIDs • Contains all applications from previous result
4	UPnP Cert App Listing Filter Single match	Invoke Get Certified Applications List action • AppCertFilter set to include name with an entry available from the list in step 2. • ProfileID set to 0	• Receive comma-separated list of appIDs • Contains all matching applications
5	UPnP Cert App Listing Filter Dual math	Invoke Get Certified Applications List action	• Receive comma-separated list of appIDs

Step	Name	Description	Expected Result
		• AppCertFilter set to include name and restricted with entries available from the list in step 2. • ProfileID set to 0	• Contains all matching application
6	UPnP Server Disconnect	See definitions in [4]	

Table 14: UPnP AppCertFilter – Test Steps

3.4.3 SR/UPNP/CERT/AppCertFilterEntityNames

Requirement: MANDATORY

Condition: None

Test the functionality of the `AppCertFilter`. The test case validates the correct behavior of the `AppCertFilter` with respect to the certifying entity values. The entries selected for the test case, are commonly used ones from MirrorLink Client devices. This test case requires the following test applications to be installed on the device.

- App 1: CCC-Drive certified for EU
- App 2: CCC-Drive certified for EU,USA
- App 3: CCC-Drive certified for JPN
- App 4: CCC-Drive certified for CHN
- App 5: CCC-Drive certified for EU,USA,JPN,CHN
- App 6: CCC-Base certified for all locales
- App 7: CTS member certified

All applications MUST be installed on the DUT prior starting the test. All applications MUST have validated application certificates.

Step	Name	Description	Expected Result
1	UPnP Server Connect	See definitions in [4] CTS uses "`CTS`" as the manufacturer name in the SetClientProfile action.	
2	UPnP Action GetCertifiedApplicationList	Invoke Get Certified Applications List action • Empty `AppCertFilter` • `ProfileID` set to 0	• Receive valid XML, formatted according to the specification
3	UPnP Cert App Listing Filter CCC	Invoke Get Certified Applications List action • `AppCertFilter` set to "`name="CCC"`" • `ProfileID` set to 0	• Receive comma-separated list of appIDs • Contains all CCC certified applications • Apps 1, 2, 3, 4, 5, 6 included • App 7 excluded
4	UPnP Cert App Listing Filter CCC Drive EU	Invoke Get Certified Applications List action • `AppCertFilter` set to "`name="CCC", restricted="EU"`" • `ProfileID` set to 0	• Receive comma-separated list of appIDs • Contains all CCC drive-certified applications for EU • Apps 1, 2, 5 included • Apps 3, 4, 6, 7 excluded

Step	Name	Description	Expected Result
5	UPnP Cert App Listing Filter CCC Drive USA	Invoke Get Certified Applications List action • `AppCertFilter` set to `“name="CCC", restricted="USA"”` • `ProfileID` set to 0	• Receive comma-separated list of appIDs • Contains all CCC drive-certified applications for USA • Apps 2, 5 included • Apps 1, 3, 4, 6, 7 excluded
6	UPnP Cert App Listing Filter CCC Drive JPN	Invoke Get Certified Applications List action • `AppCertFilter` set to `“name="CCC", restricted="JPN"”` • `ProfileID` set to 0	• Receive comma-separated list of appIDs • Contains all CCC drive-certified applications for Japan • Apps 3, 5 included • Apps 1, 2, 4, 6, 7 excluded
7	UPnP Cert App Listing Filter CCC Drive CHN	Invoke Get Certified Applications List action • `AppCertFilter` set to `“name="CCC", restricted="CHN"”` • `ProfileID` set to 0	• Receive comma-separated list of appIDs • Contains all CCC drive-certified applications for China • Apps 4, 5 included • Apps 1, 2, 3, 6, 7 excluded
8	UPnP Cert App Listing Filter unknown locale	Invoke Get Certified Applications List action • `AppCertFilter` set to `“name="CCC", restricted="ABC"”` • `ProfileID` set to 0	• Receive empty list of appIDs
9	UPnP Cert App Listing Filter Member	Invoke Get Certified Applications List action • `AppCertFilter` set to `“name="CTS"”` • `ProfileID` set to 0	• Receive comma-separated list of appIDs • Contains all CTS Member certified applications. • App 7 included • Apps 1, 2, 3, 4, 5, 6 excluded
10	UPnP Cert App Listing Filter all apps	Invoke Get Certified Applications List action • `AppCertFilter` set to `“*”` • `ProfileID` set to 0	• Receive comma-separated list of appIDs • Contains all certified applications • Apps 1, 2, 3, 4, 5, 6, 7 included
11	UPnP Server Disconnect	See definitions in [4]	

Table 15: UPnP AppCertFilter – Certifying Entities

3.4.4 SR/UPNP/CERT/GetApplicationCertificateInfo

Requirement: MANDATORY

Condition: None

Invoking the Get Application Certificate Info action and checking for validity on the returned xml.

Step	Name	Description	Expected Result
1	UPnP Server Connect	See definitions in [4]	
2	UPnP Action GetApplicationList	Invoking Get Application List action • Empty AppListingFilter • ProfileID set to 0	• Receive valid XML
3	UPnP Action GetCertifiedApplicationList	Invoke Get Certified Applications List action • Empty AppCertFilter • ProfileID set to 0	• Receive comma-separated list of appIDs
4	UPnP Get Application Certification Info	Invoke Get Application Certificate Info action for all applications from previous step.	• Receive valid XML, formatted according to the specification • Contains valid XML Signature element
5	Check App Certificate Values	CTS downloads the application certificates from the provided `appCertificateURL`. Note: The CTS MUST exclude developer application for executing this test step.	• Certificate of base or drive certified apps can be downloaded • Entries from the app certificate match the entries in the Application Certificate Info
6	UPnP Server Disconnect	See definitions in [4]	

Table 16: UPnP Action GetApplicationCertificateInfo – Test Steps

3.4.5 SR/UPNP/CERT/GetAppCertificationStatusTrue

Requirement: MANDATORY

Condition: None

Invoking the Get App Certification Status action and checking for validity of the response.

Step	Name	Description	Expected Result
1	UPnP Server Connect	See definitions in [4]	
2	UPnP Action GetCertifiedApplicationList	Invoke Get Certified Applications List action • Empty AppCertFilter • ProfileID set to 0	• Receive comma-separated list of appIDs
3	UPnP Get App Certification Status	Invoke Get App Certification Status action for all applications from previous step. • Empty AppCertFilter • ProfileID set to 0	• Receive true response
4	UPnP Server Disconnect	See definitions in [4]	

Table 17: UPnP Action GetAppCertificationStatus (true) – Test Steps

3.4.6 SR/UPNP/CERT/GetAppCertificationStatusFalse

Requirement: MANDATORY

Condition: None

Invoking the Get App Certification Status action and checking for validity of the response.

Step	Name	Description	Expected Result
1	UPnP Server Connect	See definitions in [4]	
2	UPnP Action GetApplication-List	Invoke Get Application List action • Empty AppListingFilter • ProfileID set to 0	• Receive comma-separated list of appIDs
3	UPnP Action GetCertifiedAp-plicationList	Invoke Get Certified Applications List action • Empty AppCertFilter • ProfileID set to 0	• Receive valid XML, formatted according to the specification
4	UPnP Get App Certification Status	Invoke Get App Certification Status action for all applications listed in response to step 2, which are not included in the response from step 3. • Empty AppCertFilter • ProfileID set to 0	• Receive false response
5	UPnP Server Disconnect	See definitions in [4]	

Table 18: UPnP Action GetAppCertificationStatus (false) – Test Steps

3.5 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.

3.5.1 SR/UPNP/APP/PICS/ServiceXml

Requirement: MANDATORY

Condition: None

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

Step	Name	Description	Expected Result
1	UPnP Server Connect	See Definitions	
2	Check PICS feature	FEAT_SERVER_UPNP_AppStatusUpdate	• `AppStatusUpdate` listed as evented variable in Application Server Service description.
3	Check PICS feature	FEAT_SERVER_UPNP_AppListUpdate	• `AppListUpdate` listed as evented variable in Application Server Service description.
4	Check PICS feature	FEAT_SERVER_UPnP_GetApplicationList	• `GetApplicationList` listed as action in Application Server Service description.

Step	Name	Description	Expected Result
5	Check PICS feature	FEAT_SERVER_UPnP_LaunchApplication	• LaunchApplication listed as action in Application Server Service description.
6	Check PICS feature	FEAT_SERVER_UPnP_GetApplicationStatus	• GetApplicationStatus listed as action in Application Server Service description.
7	Check PICS feature	FEAT_SERVER_UPnP_TerminateApplication	• TerminateApplication listed as action in Application Server Service description.
8	Check PICS feature	FEAT_SERVER_UPnP_GetCertifiedApplicationsList	• GetCertifiedApplicationList listed as action in Application Server Service description.
9	Check PICS feature	FEAT_SERVER_UPnP_GetAppCertificationStatus	• GetAppCertificationStatus listed as action in Application Server Service description.
10	Check PICS feature	FEAT_SERVER_UPnP_GetApplicationCertificateInfo	• GetApplicationCertificateInfo listed as action in Application Server Service description.

1 Table 19: MirrorLink Server Application Server Service XML settings PICS Checkup

2

4 CLIENT FEATURE TEST CASES

4.1 Application Server Services

4.1.1 CL/UPNP/APP/GetApplicationList

Requirement: MANDATORY

Condition: None

Tests if the UPnP Control Point is requesting application listing.

Step	Name	Description	Expected Result
1	UPnP Control Point Connect	See definitions in [4]	
2	UPnP Action Get Application List	No further action required	• Invoke Get Application List action

Table 20: UPnP Get Application List

4.1.2 CL/UPNP/APP/LaunchApplication

Requirement: MANDATORY

Condition: None

This test asks the user to launch an application via the client and waits for the UPnP action.

Step	Name	Description	Expected Result
1	UPnP Control Point Connect	See definitions in [4]	
2	UPnP Action Get Application List	No further action required	• Invoke Get Application List action
3	UPnP Action Launch Application	User is asked to launch an application via the client and waits for the UPnP event	• Invoke Launch Application action • AppID is known

Table 21: UPnP Action Launch Application – Test Steps

4.1.3 CL/UPNP/APP/TerminateApplication

Requirement: CONDITIONAL

Condition: Client support UPnP Terminate Application

This test starts an application and asks the user to terminate an application via the client and waits for the UPnP action.

Step	Name	Description	Expected Result
1	UPnP Control Point Connect	See definitions in [4]	
2	UPnP Action Get Application List	No further action required	• Invoke Get Application List action
3	UPnP Action Launch Application	User is asked to launch an application via the client and waits for the UPnP event.	• Invoke Launch Application action • AppID is known

Step	Name	Description	Expected Result
4	UPnP Action Terminate Application	User is asked to terminate the same application via the client and waits for the UPnP event.	• Invoke Launch Application action • AppID is same as above

Table 22: UPnP Action Terminate Application – Test Steps

4.1.4 CL/UPNP/APP/GetApplicationStatus

Requirement: CONDITIONAL

Condition: Client support UPnP Get Application Status

The tests checks if the client request new Application Status.

Step	Name	Description	Expected Result
1	UPnP Control Point Connect	See definitions in [4]	
2	UPnP Action Get Application List	No further action required	• Invoke Get Application List action
3	UPnP Action GetApplication-Status	User is asked to execute the know steps to trigger a Get Application Status action.	• Invoke Get Application Status • AppID is known

Table 23: UPnP Action GetApplicationStatus– Test Steps

4.1.5 CL/UPNP/APP/SignatureSuccess

Requirement: CONDITIONAL

Condition: Verification of XML Signature

Tests if the UPnP Control Point verifies the XML signature.

Step	Name	Description	Expected Result
1	UPnP Control Point Connect	See definitions in [4]	
2	DAP Client Launch	See Definitions in [3] Provide Application Listing with correct XML signature value	
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) • DUT shows list of applications

Table 24: Application Listing – Signature Success

4.1.6 CL/UPNP/APP/SignatureWrong

Requirement: CONDITIONAL

Condition: Verification of XML Signature

Tests if the UPnP Control Point verifies the XML signature.

Step	Name	Description	Expected Result
1	UPnP Control Point Connect	See definitions in [4]	
2	DAP Client Launch	See Definitions in [3] Provide Application Listing with incorrect XML signature value	
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) • DUT does not show any application

Table 25: Application Listing – Signature Failure

4.2 Application Server Events

4.2.1 CL/UPNP/APP/AppStatusUpdateSubscribe

Requirement: CONDITIONAL

Condition: Client support UPnP App Status Update event

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

Step	Name	Description	Expected Result
1	UPnP Control Point Connect	See definitions in [4]	
2	Subscribe to App Status Update event	User is asked to execute known steps to start eventing	• Receive event subscription for App Status Update

Table 26: UPnP AppStatusUpdate Subscriptoin

4.2.2 CL/UPNP/APP/AppStatusUpdateEvent

Requirement: CONDITIONAL

Condition: Client support UPnP App Status Update event

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

Step	Name	Description	Expected Result
1	UPnP Control Point Connect	See definitions in [4]	
2	Subscribe to App Status Update event	User is asked to execute known steps to start eventing	• Receive event subscription for App Status Update
3	Receive Update Event	User is asked to execute the necessary steps, to make the MirrorLink Client react on UPnP App Status Update events (PIXIT) Send Update event for all advertised applications.	• Receive UPnP GetStatus action (may be only a subset of all appIDs)

Table 27: UPnP AppStatusUpdate Event

If this test passes, the, CL/UPNP/APP/AppStatusUpdateSubscribe test cases passes as well.

4.2.3 CL/UPNP/APP/AppListUpdateSubscribe

Requirement: CONDITIONAL

Condition: Client support UPnP App List Update event

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

Step	Name	Description	Expected Result
1	UPnP Control Point Connect	See definitions in [4]	
2	Subscribe to App List Update event	User is asked to execute known steps to start eventing	• Receive event subscription for App List Update

Table 28: UPnP AppListUpdate Subscription

4.2.4 CL/UPNP/APP/AppListUpdateEvent

Requirement: CONDITIONAL

Condition: Client supports UPnP App List Update event

Test if the UPnP Control Point reacts on an UPnP App List Update event

Step	Name	Description	Expected Result
1	UPnP Control Point Connect	See definitions in [4]	
2	Subscribe to App List Update event	User is asked to execute known steps to start eventing	• Receive event subscription for App List Update
3	Receive Update Event	User is asked to execute the necessary steps, to make the MirrorLink Client react on UPnP App List Update events (PIXIT) Send Update event for all advertised applications	• Receive UPnP GetApplicationList action (may be for individual appIDs)

Table 29: UPnP AppListUpdate Event

If this test passes, the, CL/UPNP/APP/AppListUpdateSubscribe test cases passes as well.

4.3 Certification

4.3.1 CL/UPNP/CERT/GetCertifiedApplicationsList

Requirement: CONDITIONAL

Condition: Support for GetCertifiedApplicationsList action

Tests if the UPnP Control Point is requesting certified application listing.

Step	Name	Description	Expected Result
1	UPnP Control Point Connect	See definitions in [4]	
2	UPnP Action Get Certified Applications List	Test Engineer MAY execute steps to trigger this action on the DUT	• Invoke Get Certified Applications List action

Table 30: UPnP Action GetCertifiedApplicationsList – Test Steps

4.3.2 CL/UPNP/CERT/GetAppCertificationStatus

Requirement: CONDITIONAL

Condition: Support for GetAppCertificationStatus action

Tests if the UPnP Control Point is requesting application certification status.

Step	Name	Description	Expected Result
1	UPnP Control Point Connect	See definitions in [4]	
2	UPnP Action Get App Certification Status	Test Engineer MAY execute steps to trigger this action on the DUT	• Invoke Get App Certification Status action

Table 31: UPnP Action GetAppCertificationStatus – Test Steps

4.3.3 CL/UPNP/CERT/GetApplicationCertificateInfo

Requirement: CONDITIONAL

Condition: Support for GetApplicationCertificateInfo action

Tests if the UPnP Control Point is requesting application certificate info.

Step	Name	Description	Expected Result
1	UPnP Control Point Connect	See definitions in [4]	
2	UPnP Action Get Application Certificate Info	Test Engineer MAY execute steps to trigger this action on the DUT	• Invoke Get Application Certificate Info action

Table 32: UPnP Action GetApplicationCertificateInfo – Test Steps

4.3.4 CL/UPNP/CERT/SignatureSuccess

Requirement: CONDITIONAL

Condition: Support for GetApplicationCertificateInfo action AND
Support for XML Signature Verification

Tests if the UPnP Control Point verifies the XML signature.

Step	Name	Description	Expected Result
1	UPnP Control Point Connect	See definitions in [4]	
2	DAP Client Launch	See Definitions in [3]	
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)
5	UPnP Action Get Application Certificate Info	Test Engineer MAY execute steps to trigger this action on the DUT	• Invoke Get Application Certificate Info action
6	Certificate Info	Provide Certificate Info with correct XML signature value	• DUT includes application into list of certified applications (PIXIT)

Table 33: Certificate Information – Signature Success

4.3.5 CL/UPNP/CERT/SignatureWrong

Requirement: CONDITIONAL

Condition: Support for GetApplicationCertificateInfo action AND
Support for XML Signature Verification

Tests if the UPnP Control Point verifies the XML signature.

Step	Name	Description	Expected Result
1	UPnP Control Point Connect	See definitions in [4]	
2	DAP Client Launch	See Definitions in [3]	
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)
5	UPnP Action Get Application Certificate Info	Test Engineer MAY execute steps to trigger this action on the DUT	• Invoke Get Application Certificate Info action
6	Certificate Info	Provide Certificate Info with incorrect XML signature value	• DUT does not include application into list of certified applications (PIXIT)

Table 34: Certificate Information – Signature Failure

4.4 Performance

4.4.1 CL/UPNP/PERF/APP/SoapTimeout

Requirement: MANDATORY

Condition: None

This test validates, whether the DUT can handle a MirrorLink Server device, responding slowly to the Application Server service's UPnP actions.

Step	Name	Description	Expected Result
1	UPnP Control Point Connect	See definitions in [4]	
2	UPnP App Server Service timeout	Test Engineer is asked to launch all listed applications via the client (i.e. launch one app, check the user interface and then launch the next app). Test Engineer may need to do manual steps to launch DAP, CDB and RTP endpoints from the DUT. CTS provides response to all received Application Server service's SOAP action at 90% of the SOAP' timeout.	• User interface of all launched applications is seen on the HU screen. • DAP endpoint launched and connected • RTP Server launched and connected • CDB endpoint launched and connected (if supported)

Table 35: UPnP App Server Service - SOAP timeout

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 Application Server Service”, Version 1.1, CCC-TS-024

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

[4] Car Connectivity Consortium, “MirrorLink – UPnP Server Device Test Specification”, Version 1.1; CCC-TS-028