Car Connectivity Consortium

MirrorLink®

MirrorLink Over Wi-Fi Display Test Specification

Version 1.2.2
(CCC-TS-067)

VERSION HISTORY

Version	Date	Comment
1.2.0	18 June 2014	Approved Version
1.2.1	10 November 2014	Approved Errata Version
1.2.2	10 June 2015	Approved Errata Version

LIST OF CONTRIBUTORS

Brakensiek, Jörg Microsoft Corporation
Kafle, Padam Qualcomm
Subramaniam, Vijay Qualcomm

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

GO	Group Owner
HDCP	High-bandwidth Digital Content Protection
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
Miracast	Commercial denomination of WFD
ML	MirrorLink
OUI	Organizationally Unique Identifier
P2P	Peer to Peer Link
Sink Device	A device that receives multimedia content from a WFD source over a Wi-Fi link and renders it.
Source Device	A device that supports streaming multimedia content to a WFD sink(s) over a Wi-Fi link.
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UIBC	User Input Back Channel
UPnP	Universal Plug and Play
USB	Universal Serial Bus
VNC	Virtual Network Computing
WFD	Wi-Fi Display
WPS	Wi-Fi Protected Setup

1 ABOUT

This document specifies MirrorLink protocol conformance test cases for the MirrorLink over Wi-Fi Display 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

For the scope of this test specification, the MirrorLink Server refers to the Wi-Fi Display Source device and the MirrorLink Client refers to the Wi-Fi Display Primary Sink device, which are defined in [3]. When the term "Sink" is used in rest of this test specification, it refers to a WFD Primary Sink device.

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

This test specification MUST be applicable to MirrorLink Client devices, implementing a WFD Sink, and to MirrorLink Server devices, implementing a WFD Source. The applicable WFD Source and Sink functionality MUST be certified through Wi-Fi Alliance's Miracast certification program [5]. Details are described in the MirrorLink device certification program management document [11].

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

The following definitions are frequently used in different server test cases. Usage is indicated by the given designator name.

2.2.1 WFD Connection Setup

The MirrorLink Server sets up a WFD connection by establishing Wi-Fi P2P link to a MirrorLink Client. The alternative method of establishing WFD connection using TDLS [3] is OPTIONAL, and not part of this test specification.

This definition contains all necessary steps to complete the Wi-Fi P2P connection with WFD device discovery parameter exchange.

Step	Name	Description	Expected Result
1	Power on WLAN radio	Switch on WLAN radio. Start ML Server and the reference ML Client (if not done automatically).	Wi-Fi on MirrorLink Server is detectable.
2	Enable Wi-Fi Display Functionality	Switch on the MirrorLink over Wi-Fi Display by using the application or user input If a persistent WFD group exists between the source and sink, this step is not required.	

Step	Name	Description	Expected Result
3	Wi-Fi P2P connection	Establish Wi-Fi Peer to Peer connection between MirrorLink Server to the reference MirrorLink Client WPS process as needed ML Server MAY have P2P Group Owner or Client role	• P2P Device Discovery happens with the WFD IE and CCC IE included • Wi-Fi P2P connection established as per [6]. P2P Group Owner or Client role agreed

2.2.2 WFD Session Setup

The MirrorLink Server starts a WFD session when the MirrorLink over Wi-Fi Display is switched on, and subsequently an application over WFD is launched as described in [2].

Step	Name	Description	Expected Result
1	WFD Connection Setup	See Definitions	• The WFD connection is setup with a P2P link to the WFD capable reference Client device
2	UPnP Server Connect	CTS accesses the DUT's Device XML at the URL derived from the CCC IE. Validate that Device and Service XMLs have correct XML format, includes required service types and their control and event URLs.	• Valid device description (according to specification) • Support for TmApplicationServer:1 service • Support for TmClientProfile:1 service
3	UPnP advertisements	DUT starts advertising using SSDP:alive	• DUT sends SSDP:alive messages • URL is identical to the one derived from the CCC IE.
4	MirrorLink Session Setup	See [8]	• The reference Client sets its Client Profile and follows the MirrorLink session setup sequence as per [8].
5	WFD TCP Connection Setup	WFD TCP connection setup starts once the step 1 has been completed. This step and steps 6-7 may occur in parallel to steps 2-4 above.	• The TCP connection is setup to the reference Client, and RTSP stack is active
6	WFD Capability Negotiation	Complete RTSP M1-M4 RTSP Message Exchange as described in section 4.6 of [3].	
7	WFD Session Establishment	Complete M5-M7 RTSP Message Exchange as described in section 4.8 of [3]. The reference Client (CTS) may send a RTSP M9 request (PAUSE) to pause the streaming of audio and/or video if step 4 has not completed by end of M7 request/response.	• The WFD session setup is completed with RTSP OK in M7 Response sent from the WFD Source • UIBC session is activated

Step	Name	Description	Expected Result
8	First application launch over WFD	Launch an application over WFD via the UPnP Launch Application. User input may be required to trigger the application. If the streaming of audio and/or video has been paused in step 7, the reference Client MUST send the M7 request (PLAY) to resume A/V streaming once the first application over WFD is launched.	• Audio/video streaming starts so that the reference Client shows the display from the launched application over WFD.

2.2.3 WFD Session Teardown

The MirrorLink Server tears down the WFD session when the Client sends RTSP tear down command, which may be caused when the MirrorLink session has been terminated.

Step	Name	Description	Expected Result
1	Trigger Teardown of WFD session	DUT triggers the teardown of the WFD session. CTS sends RTSP TEARDOWN (M5 Response) message with Status OK. This step may be skipped if DUT autonomously sends RTSP TEARDOWN message (step 2))	• DUT sends RTSP TEARDOWN (M5 Request) Message.
2	Teardown of WFD session	CTS sends RTSP TEARDOWN (M8 Request) Message to DUT	• DUT sends RTSP TEARDOWN (M8 Response) Message with Status OK. • If DUT is the P2P GO then it may send a de-authentication message to the Sink and then terminate the P2P Group

2.2.4 MirrorLink Session Setup

The MirrorLink Server established the MirrorLink session.

Step	Name	Description	Expected Result
1	Client Profile	CTS sends the following UPnP actions: • Get Max Num Profiles • Set Client Profile	• Receive Client Profile response • No change to the presentations elements. • All responses sent over Wi-Fi
2	Application Listing	CTS sends the following UPnP actions: • Get Application List	• Receive Application List response • Protocol identifier of regular applications given with “WFD” • Response sent over Wi-Fi

Step	Name	Description	Expected Result

3	DAP	CTS sends the following UPnP actions:
		• Launch Application (DAP)
		• DAP protocol execution
		• Terminate Application (DAP)
			• DAP endpoint launched
			• DAP protocol executed;
			  ○ may not include WFD end-point
			  ○ must include device and UPnP server
			• DAP endpoint terminated
			• All responses sent over Wi-Fi

4	CDB	CTS sends the following UPnP actions: • Launch Application (CDB), if available	• CDB endpoint launched • Response sent over Wi-Fi
5	Application List	CTS retrieves application icons and presents available applications.	• List of applications available • Application advertised with "WFD" remote protocol identifier.

2.3 Client Definitions

The following definitions are frequently used in different client test cases. Usage is indicated by the given designator name.

2.3.1 WFD Connection Setup

The MirrorLink Client sets up a WFD connection by establishing Wi-Fi P2P link to a MirrorLink Server. The alternative method of establishing WFD connection using TDLS [3] is OPTIONAL, and not part of this test specification.

This definition contains all necessary steps to complete the WFD connection.

Step	Name	Description	Expected Result
1	Power on WLAN radio	Switch on WLAN radio. Start the ML Client and the reference ML Server (if not done automatically).	• Wi-Fi on MirrorLink Client is detectable.
2	Enable Wi-Fi Display Functionality	Switch on the MirrorLink over Wi-Fi Display by using an application or user input If a persistent WFD group exists between the source and sink, this step is not required.	
3	Wi-Fi P2P Connection Setup	Establish Wi-Fi Peer to Peer connection between MirrorLink Client to the reference MirrorLink Server WPS process as needed ML Client MAY have P2P Group Owner or Client role	• P2P Device Discovery happens with the WFD IE and CCC IE included • Wi-Fi P2P connection established as per [6]. P2P Group Owner or Client role agreed

2.3.2 WFD Session Setup

The MirrorLink Client starts a WFD session when the MirrorLink over Wi-Fi Display is switched on, and subsequently an application over WFD is launched as described in [2].

Step	Name	Description	Expected Result
1	WFD Connection Setup	See Definitions	• The WFD connection is setup with a P2P link to the WFD capable reference Server device
2	UPnP Control Point Connect	DUT accesses the DUT's Device XML. CTS starts sending SSDP:alive messages only AFTER the DUT has retrieved the Device XML.	• The DUT retrieves the Device XML from the URL provided in the CCC IE or SSDP:discover response • DUT may retrieve the Service XML. • DUT may send `SSDP:discover` message (must be formatted correctly)

Step	Name	Description	Expected Result
3	MirrorLink Session Setup	See [8]	• The Client sets its Client Profile and follows the MirrorLink session setup sequence as per [8].
4	WFD TCP Connection Setup	WFD TCP connection setup starts once the step 1 has been completed. This step may occur in parallel to steps 2-3 above.	• The TCP connection is setup to the reference Server, and RTSP stack is active
5	WFD Capability Negotiation	Complete RTSP M1-M4 RTSP Message Exchange as described in section 4.6 of [3].	
6	WFD Session Establishment	Complete M5-M7 RTSP Message Exchange as described in section 4.8 of [3]. Send a RTSP M9 request (PAUSE) to pause the streaming of audio and/or video if needed (step 3 has not completed by end of M7 request/response and Client is set in drive mode).	• The WFD session setup is completed with RTSP OK in M7 Response sent from the reference Server • UIBC session is activated
7	First application launch over WFD	Launch an application over WFD via the UPnP Launch Application. User input may be required to trigger the application. If the streaming of audio and/or video has been paused in step 7, the Client MUST send the M7 request (PLAY) to resume A/V streaming once the first application over WFD is launched.	• Audio/video streaming starts so that the reference Client shows the display from the launched application over WFD.

2.3.3 WFD Session Teardown

The MirrorLink Client tears down the WFD session when the Client sends RTSP tear down command, which may be caused when the MirrorLink session has been terminated.

Step	Name	Description	Expected Result
1	Trigger Teardown of WFD session	CTS sends RTSP TEARDOWN (M5 Request) Message to DUT	• DUT sends RTSP TEARDOWN (M5 Response) Message with Status OK.
2	Teardown of WFD session	DUT is tearing down WFD session. CTS sends RTPS TEARDOWN (M8 Response) message with Status OK.	• DUT sends RTSP TEARDOWN (M8 Request) Message. • If the Sink is the P2P GO then it may send a de-authentication message to the Source and then terminate the P2P Group

2.3.4 MirrorLink Session Setup

The MirrorLink Client established the MirrorLink session.

Step	Name	Description	Expected Result
1	Client Profile	CTS receives at least the following UPnP action: • Set Client Profile The DUT may send additional UPnP Client Profile actions: • Event registration • Get Client Profile • Get Max Num Profiles	• Client Profile includes VNCU and WFD • All actions sent over Wi-Fi
2	Application Listing	CTS receives at least the following UPnP actions: • Get Application List The DUT may send additional UPnP Application Server Service actions: • Event registration • Get Application Status • Application certificate related actions	• All actions sent over Wi-Fi
3	DAP	The DUT may send additional UPnP Application Server Service actions: • Launch Application (DAP) • Terminate Application (DAP) The DUT may initiate and execute other protocols: • DAP	• All actions sent over Wi-Fi
4	CDB	The DUT may send additional UPnP Application Server Service actions: • Launch Application (CDB) The DUT may initiate and execute other protocols: • CDB	• All actions sent over Wi-Fi
5	Application List	DUT must present a list of available applications to the user.	• List of applications available.

3 SERVER FEATURE TEST CASES

3.1 WFD Device Discovery at MirrorLink Server

These tests verify that the MirrorLink Server is capable to perform WFD Device Discovery while operating as a WFD Source, and sends CCC IE [3] with CCC specific OUI and the MirrorLink UPnP Device Information during the WFD device discovery phase. All other steps for verification of WFD Device Discovery are considered verified during the WFA certification as part of [5].

3.1.1 SR/WFD/DISCOVERY/ProbeRequestDuringScanState

Requirement: MANDATORY

Condition: None

Tools: Sniffer supporting Wi-Fi Display protocol including the required specifications it uses.

This test checks that the MirrorLink server device advertises its capabilities including the CCC IE with CCC specific OUI and the MirorLink UPnP Device Information in the probe request frames it sends during the WFD Device Discovery at Scan state.

Step	Name	Description	Expected Result
1	Activate MirrorLink over Wi-Fi Display	Switch on WLAN radio Switch on the MirrorLink over Wi-Fi Display by using an application or user input (if not done automatically or from previous tests)	• The Server device initiates the WFD connection setup by starting the WFD discovery
2	Setting into Scan state	Put the ML Server device in Scan State with channel 6 as listen channel See Vendor Instructions for setup	• Scan covers all supported channels (both in 2.4 GHz and 5 GHz)
3	Sending Probe Requests	Start sending Probe Request frames Capture the Probe Request frames using the Sniffer Test that the CCC IE with correct information is sent in the Probe Request frames including in a non-social channel.	• CCC IE is present with OUI value set to 04 DF 69 (CCC), and the MirrorLink UPnP Device Information sub element set with the values configured for the device. The Internet Accessibility sub element may be present in the CCC IE if supported (optional). • Verify that UPnP service related information (e.g. UPnP App Server Service, Client Profile Service, Notification Server Service) are filled correctly according to the PICS for the MirrorLink Server. • The WFD device information in the WFD IE may be verified to confirm that it is set as a “Source” and the WFD Session Availability set as 0b01 “available”

Step	Name	Description	Expected Result
4	WLAN disconnection	Power off WLAN radio at the MirrorLink Server	

3.1.2 SR/WFD/DISCOVERY/ProbeRequestDuringFindState

Requirement: MANDATORY

Condition: None

Tools: Sniffer supporting Wi-Fi Display protocol including the required specifications it uses

This test checks that the MirrorLink server device advertises its capabilities including the CCC IE with CCC specific OUI and the MirorLink UPnP Device Information in the probe request frame it sends during the WFD Device Discovery at Find state.

Step	Name	Description	Expected Result
1	Activate MirrorLink over Wi-Fi Display	Switch on WLAN radio Switch on the MirrorLink over Wi-Fi Display by using an application or user input (if not done automatically or from previous tests)	• The Server device initiates the WFD connection phase by starting the WFD discovery
2	Setting into Find state	Put the ML Server device in device discovery phase at Find state with channel 1 as the listen channel See Vendor Instructions for setup	
3	Sending Probe Requests	Start sending Probe Request frames Capture the Probe Request frames using the Sniffer Test that the CCC IE with correct information is sent in the Probe Request frames in a social channel.	• CCC IE is present with OUI value set to 04 DF 69 (CCC), and the MirrorLink UPnP Device Information sub element set with the values configured for the device. The Internet Accessibility sub element may be present in the CCC IE if supported (optional). • Verify that UPnP service related information (e.g. UPnP App Server Service, Client Profile Service, Notification Server Service) are filled correctly according to the PICS for the MirrorLink Server. • The WFD device information in the WFD IE may be verified to confirm that it is set as a “Source” and the WFD Session Availability set as 0b01 “available”
4	WLAN disconnection	Power off WLAN radio at the MirrorLink Server	

3.1.3 SR/WFD/DISCOVERY/ProbeResponseDuringListenState

Requirement: MANDATORY

Condition: None

Testing Tools: Sniffer supporting Wi-Fi Display protocol, Packet Injector tool or a reference WFD capable MirrorLink Client device.

This test checks that the MirrorLink server device includes its capabilities including the CCC IE with CCC specific OUI and the MirorLink UPnP Device Information in the probe response frame it sends during the WFD Device Discovery at Listen state.

Step	Name	Description	Expected Result
1	Activate MirrorLink over Wi-Fi Display	Switch on WLAN radio Switch on the MirrorLink over Wi-Fi Display by using an application or user input (if not done automatically or from previous tests)	• The Server device initiates the WFD connection phase by starting the WFD discovery
2	Setting into Listen state	Put the ML Server device in device discovery phase at Listen state with channel 1 as listen channel See Vendor Instructions for setup	
3	Sending Test Probe Requests	Use a Packet Injector tool or a test WFD device to send several Probe Request Frames with WFD and CCC IEs, to the destination address set as broadcast address. Capture the Probe Request frames using the Sniffer	
4	Sending Probe Responses	Start sending Probe Response frames Capture the Probe Response frames using the Sniffer Test that the CCC IE with correct information is sent in the Probe Response frames including in a non-social channel.	• Probe Response from the Server device within 1 second • CCC IE is present with OUI value set to 04 DF 69 (CCC), and the MirrorLink UPnP Device Information sub element set with the values configured for the device. The Internet Accessibility sub element may be present in the CCC IE if supported (optional). • Verify that UPnP service related information (e.g. UPnP App Server Service, Client Profile Service, Notification Server Service) are filled correctly according to the PICS for the MirrorLink Server. • The WFD device information in the WFD IE may be verified to confirm that it is set as a “Source” and the WFD Session Availability set as 0b01 “available”
5	WLAN disconnection	Power off WLAN radio at the MirrorLink Server	

3.1.4 SR/WFD/DISCOVERY/BeaconAndProbeResponseInGOMode

Requirement: MANDATORY

Condition: None

Testing Tools: Sniffer supporting Wi-Fi Display protocol, Packet Injector tool or a reference WFD capable MirrorLink Client device.

This test checks that the MirrorLink server device includes its capabilities including the CCC IE with CCC specific OUI and the MirorLink UPnP Device Information in the beacon and probe response frames it sends to support the WFD Device Discovery while operating as an autonomous Group Owner.

Step	Name	Description	Expected Result
1	Setting as Autonomous Group Owner	Switch on WLAN radio Put the ML Server device into the autonomous Group Owner mode with channel 11 as its operating channel and listen channel See Vendor Instructions for setup	
2	Sending Test Probe Requests	Use a Packet Injector tool or a test WFD device to send several Probe Request Frames with WFD and CCC IEs, to the destination address set as broadcast address. Capture the Probe Request frames using the Sniffer	
3	Sending Probe Responses	Start sending Probe Response frames Capture the Probe Response frames using the Sniffer Test that the CCC IE with correct information is sent in the Probe Response frames including in a non-social channel.	• Probe Response from the Server device within 1 second • CCC IE is present with OUI value set to 04 DF 69 (CCC), and the MirrorLink UPnP Device Information sub element set with the values configured for the device. The Internet Accessibility sub element may be present in the CCC IE if supported (optional). • Verify that UPnP service related information (e.g. UPnP App Server Service, Client Profile Service, Notification Server Service) are filled correctly according to the PICS for the MirrorLink Server. • The WFD device information in the WFD IE may be verified to confirm that it is set as a “Source” and the WFD Session Availability set as 0b01 “available”
4	Verification of Beacon Frames	Capture the Beacon frames transmitted by the Server device using the Sniffer	• CCC IE is present with OUI value set to 04 DF 69 (CCC), and the MirrorLink UPnP Device Information sub element set with the

Step	Name	Description	Expected Result
		Test that the CCC IE with correct information is sent in the Beacon frames.	values configured for the device. The Internet Accessibility sub element may be present in the CCC IE if supported (optional). • Verify that UPnP service related information (e.g. UPnP App Server Service, Client Profile Service, Notification Server Service) are filled correctly according to the PICS for the MirrorLink Server. • The WFD device information in the WFD IE may be verified to confirm that it is set as a “Source” and the WFD Session Availability set as 0b01 “available”
4	WLAN disconnection	Power off WLAN radio at the MirrorLink Server	

3.2 UPnP Session Setup over WFD Connection

3.2.1 SR/WFD/UPnP/UPnPSetupAfterWFDConnection

Requirement: MANDATORY

Condition: None

This test checks if the MirrorLink Server is able to start the UPnP session based on the information (CCC IE) received during the WFD Discovery (see Section 3.1) after the WFD connection setup.

Step	Name	Description	Expected Result
1	WFD Connection Setup	See Definitions Use channel 1 as its operating channel and listen channel for 2.4 GHz Use channel 36 as operating channel if 5 GHz is supported (repeat steps 1-4)	• WFD connection is setup from the Server with P2P connection to a reference Client • The Association and 4-way handshake for authentication is captured by the Sniffer to verify the connection is completed
2	UPnP Server Connect	CTS accesses the DUT's Device XML at the URL derived from the CCC IE. Validate that Device and Service XMLs have correct XML format, includes required service types and their control and event URLs.	• Valid device description (according to specification) • Support for TmApplicationServer:1 service • Support for TmClientProfile:1 service
3	UPnP advertisements	DUT starts advertising using SSDP:alive	• DUT sends SSDP:alive messages • URL is identical to the one derived from the CCC IE.
4	MirrorLink Session Setup	See Definitions	
5	UPnP Server Disconnect	See Definitions in [7]	

3.2.2 SR/WFD/UPnP/UPnPSetupSSDPdiscover

Requirement: MANDATORY

Condition: None

This test checks if the MirrorLink Server is able to start the UPnP session based on the information received from `SSDP:discover` after the WFD connection setup.

Step	Name	Description	Expected Result
1	WFD Connection Setup	See Definitions Use channel 1 as its operating channel and listen channel for 2.4 GHz Use channel 36 as operating channel if 5 GHz is supported (repeat steps 1-4)	• WFD connection is setup from the Server with P2P connection to a reference Client • The Association and 4-way handshake for authentication is captured by the Sniffer to verify the connection is completed

Step	Name	Description	Expected Result
2	UPnP Server Discovery	CTS sends `SSDP:discover` message	• DUT correctly responds to the `SSDP:discover` message • Provided URL is identical to the URL advertised in the CCC Information element.
3	UPnP Server Connect	CTS accesses the DUT's Device XML at the URL derived from the `SSDP:discover` response message. Validate that Device and Service XMLs have correct XML format, includes required service types and their control and event URLs.	• Valid device description (according to specification) • Support for TmApplicationServer:1 service • Support for TmClientProfile:1 service
4	UPnP advertisements	DUT starts advertising using SSDP:alive	• DUT sends SSDP:alive messages • URL is identical to the one derived from the CCC IE.
5	MirrorLink Session Setup	See Definitions	
6	UPnP Server Disconnect	See Definitions in [7]	

3.3 WFD Session Setup

The WFD capability exchange and negotiation steps specific to MirrorLink implementation are tested under these test cases. Other tests to verify WFD capability negotiation including the support for the UIBC feature and WFD session establishment are covered by Wi-Fi Alliance certification testing as per [5].

3.3.1 SR/WFD/SESSION/SessionStartAfterApplicationLaunch

Requirement: MANDATORY

Condition: None

Note: If the Server supports the UI blocking with video frame skipping as described in [3], the Server device MUST also undergo the testing for the test case "Video Frame Skipping on SoUT" in [5] as part of the Wi-Fi Alliance Miracast certification.

This test checks if the MirrorLink Server can start the WFD session setup procedure and start streaming audio/video correctly when a first application over WFD is launched. The WFD capability negotiation for the MirrorLink specific display resolutions and the video frame skipping feature (if supported) by the Server are tested.

Step	Name	Description	Expected Result
1	WFD Connection Setup	See Definitions Use channel 1 as its operating channel and listen channel for 2.4 GHz Use channel 36 as operating channel if 5 GHz is supported (need to repeat steps 1-5)	• The WFD connection is setup with P2P link to the WFD capable reference Server device
2	UPnP Session Setup	See 3.2.1 (perform steps 2-4)	
3	WFD TCP Connection Setup	WFD TCP connection setup starts once the step 1 has been completed. This step may occur in parallel to step 2 above.	• The TCP connection is setup to the reference Client, and RTSP stack is active

Step	Name	Description	Expected Result
4	WFD Capability Negotiation	Complete RTSP M1-M4 RTSP Message Exchange as described in section 4.6 of [3]. If supported, messages may be sent manually Using the Sniffer, capture and analyze the RTSP messages during the WFD capability negotiation	• Verify that the reference Client supports 800x480p30 (WVGA) resolution in the M3 response, and the Server selects this resolution by including it in the `wfd-video-formats` parameter in RTSP SET_PARAMETER (M4 request) • Verify that if the Client indicates support for display resolution of 1280x720 or higher, the 1280x720p30 is also supported. In this case, verify that the Server either selects this resolution or 800x480p30 in the `wfd-video-formats` parameter in M4 request. • Visually verify that there is no clipping and aspect ratio is preserved • If the Server supports video frame skipping (included as part of Miracast certification), verify that it sets the `frame-rate-control-support` within the `wfd-video-formats` parameter with b0=0b1 and b3-b1=0b000 (infinite skipping interval) in its M4 request message.
5	WFD Session Establishment	Complete M5-M7 RTSP Message Exchange If supported, messages may be sent manually Using the Sniffer, capture and analyze the RTSP messages during the WFD session establishment. The reference Client (CTS) may send a RTSP M9 request (PAUSE) to pause the streaming of audio and/or video if step 2 has not completed by end of M7 request/response.	• Verify that the WFD session is established with RTSP OK in M7 Response sent from the WFD Source.

Step	Name	Description	Expected Result
6	First application launch over WFD	Launch an application over WFD via the UPnP Launch Application. User input may be required to trigger the application. If the streaming of audio and/or video has been paused in step 5, the reference Client MUST send the M7 request (PLAY) to resume A/V streaming once the first application over WFD is launched.	• Verify that the Audio/video streaming starts after the M7: RTSP PLAY request/response is completed, and the reference Client shows the display from the launched application over WFD.
7	WFD Session Teardown	See Definitions	

3.3.2 SR/WFD/SESSION/SupportForCCCSpecificUIBC

Requirement: MANDATORY

Condition: None

This test checks if the MirrorLink Server performs the WFD capability negotiation that include correct support for CCC specific UIBC capabilities and associated parameters.

Step	Name	Description	Expected Result
1	WFD Connection Setup	See Definitions If the WFD connection already exists, this step may be skipped	• The WFD connection is setup with the WFD capable reference Client device
2	UPnP Session Setup	See 3.2.1 (perform steps 2-3)	
3	First application launch over WFD	Launch an application over WFD via the UPnP Launch Application User input may be required to trigger the application	• The TCP connection is setup to the reference Client, and RTSP stack is active
4	WFD Capability Negotiation: M1-M2 exchange	Complete RTSP M1-M2 RTSP Message Exchange If supported, messages may be sent manually	• The RTSP messages are captured by the Sniffer to verify the parameters exchanged during WFD session establishment

Step	Name	Description	Expected Result
5	WFD Capability Negotiation: M3 request and response	Server sends RTSP GET_PARAMER request (M3) by including the `wfd_uibc_capability` parameter, and receives the M3 response with the `wfd_uibc_capability` parameter indicating the supported capabilities by the reference Client	• Verify that the Server sends the RTSP GET_PARAMER request (M3) which includes `wfd_uibc_capability` as a parameter and receives M3 response from the reference Client.
6	WFD Capability Negotiation: M4 request and response	Server device (WFD source) sends RTSP SET_PARAMER (M4) request with `wfd_uibc_capability` parameter, and receives M4 response with OK from the reference Client	• Verify that the `wfd_uibc_capability` in M4 request sent from the Server device includes correct parameters indicated to be supported by the reference Client in M3 response including the `vendor-specific-cap-info` with "04DF69" as the `vendor_OUI` and associated capability parameters that may be supported as per [2]. The ccc_event_cap_list should contain a sub-set of parameters obtained from M3 Response in Step 5. • Verify that the `wfd_uibc_capability` in M4 request sent from the Server device includes a TCP port number • Verify that if the Server includes the `wfd_uibc_setting` parameter in M4 request, then it is set to the value 'enable'.
7	RTSP M15 request and response	Skip this test if the Server included `wfd_uibc_setting` parameter in M4 request (step 6) Send the RTSP SET_PARAMER request (M15) with `wfd_uibc_setting` parameter.	• Verify that the Server includes the `wfd_uibc_setting` parameter in M15 request with its value set to 'enable'. • Receive M15 response with OK from the reference Client
8	WFD Session Establishment	Complete M5-M7 RTSP Message Exchange If supported, messages may be sent manually	• Verify that the WFD session is established • Verify that the Audio/video streaming starts after the M7: RTSP PLAY request/response is completed, and the reference Client shows the display from the launched application
9	WFD Session Teardown	See Definitions	

wfd-uibc-capability sub-parameter names	Format and Supported Values	Description
input-category-val	input_category_list=VENDOR_SPECIFIC	
vendor-specific-cap-info	vendor_specific_cap_info = OUI: 04 DF 69 ; ccc_event_cap_list = <ccc-event-inp-type>	
ccc-event-inp-type	ccc-resolution-info: Resolution/ [width pixel] [height pixel]	Sink display resolution
	ccc-display-info: Display/ [mm] [mm] [mm]	width x height x distance to user for Sink's display in mm
	ccc-keys-info: KnobKeys/ [4*HEXDIG value] DeviceKeys/ [4*HEXDIG value] MultimedeaKeys/ [4*HEXDIG value] FunctionKeys/ [2*HEXDIG value] ITUKeys/("none"/"Supported")	Lists the supported key events, see [2]
	ccc-pointer-info: Pointer/ [2*HEXDIG value]	Indicates the supported pointer event button mask, must be 0x00 00 if not supported
	ccc-touch-info: Touch/ [2*HEXDIG number value] [2*HEXDIG pressure mask value]	Indicates the number of supported simultaneous touch events (max number -1) and touch event pressure mask
	ccc-text-output: TextOutput/ [4*HEXDIG value]	Maximum length of textual meta information
	CutText/("none"/"Supported")	Include "Supported" if the sink supports Cut Text, otherwise "none" is included
tcp-port	port = <IPPORT>	Port to be used for UIBC

Table 3-1: wfd-uibc-capability parameter values supported by CCC capable WFD Source and Sink devices

3.3.3 SR/WFD/SESSION/SessionTeardownFromServer

Requirement: MANDATORY

Condition: None

This test checks if the MirrorLink Server can start the WFD session teardown when the MirrorLink over WFD session ends.

Step	Name	Description	Expected Result
1	WFD Session Setup	See Definitions If the WFD session already exists, skip this step	• Audio/video streaming starts so that the reference Client shows the display from the launched application
2	Trigger teardown of WFD session	DUT triggers the teardown of the WFD session. CTS sends RTSP SET_PARAMETER response (M5 Response) message with Status OK.	• DUT sends RTSP SET_PARAMETER request with trigger:TEARDOWN (M5 Request) Message.
3	Teardown of WFD session	CTS sends RTSP TEARDOWN (M8 Request) Message to DUT.	• DUT sends RTSP TEARDOWN (M8 Response) Message with Status OK. • If DUT is the P2P GO then it may send a de-authentication message to the Sink and then terminate the P2P Group

3.3.4 SR/WFD/SESSION/SessionTeardownFromClient

Requirement: MANDATORY

Condition: None

This test checks if the MirrorLink Client can start the WFD session teardown when the MirrorLink over WFD session ends and Server responds appropriately.

Step	Name	Description	Expected Result
1	WFD Session Setup	See Definitions If the WFD session already exists, skip this step	• Audio/video streaming starts so that the reference Client shows the display from the launched application
2	Teardown of WFD session	CTS sends RTSP TEARDOWN (M8 Request) Message to DUT.	• DUT sends RTSP TEARDOWN (M8 Response) Message with Status OK. • If DUT is the P2P GO then it may send a de-authentication message to the Sink and then terminate the P2P Group.

3.4 WFD User Inputs and Events

3.4.1 SR/WFD/USERINPUTS/KeyEvents

Requirement: MANDATORY

Condition: None

This test verifies that the MirrorLink Server correctly processes the UIBC key inputs received from the reference Client, which are CCC specific UIBC events under the Vendor Specific Input category as specified in [2]. A number of key events are sent to the Server. The testing verifies if the corresponding text is visible on the MirrorLink Server device.

This test requires a Text Test Application available on the MirrorLink Server and supported from the MirrorLink Client. If a Text Test Application is not available, X11 Key Events MUST be tested through IOP test cases.

Step	Name	Description	Expected Result
1	WFD Session Setup	See Definitions If the WFD session already exists, skip this step	• The WFD session is setup and UIBC is enabled for the CCC specific UIBC capabilities between the Server and the Client • One or more Key events supported during WFD capability negotiation (test case 3.3.2)
2	Setup for UIBC key events	Launch an application (or test application) over WFD via the UPnP Launch Application that can handle the supported key events	• The Server is ready to receive UIBC key events
3	Check X11 Key Events	Send a series of Key Press and Release events from the Client: • Letters: 'a' – 'z' • Letters: 'A' – 'Z' • Numbers: '0' – '9' • Symbols as specified: ' ', '!', '"', '#', '$', '%', '&', ''', '(', ')', '*', '+', ',', '-', '.', '/', ':', ';', '<', '=', '>', '?', '@', '[', '\', ']', '^', '_', '`', '{', '|', '}', '~', Backspace, Return	• Verify that the application behaves correctly with the key events, and the Server updates the display in accordance with the key inputs from the Client
4	Knob Key events	Launch an application (or test application) that can handle the Knob events (if not already launched) Execute all supported Knob key events (as observed in 3.3.2) one after another in the Client and verify the behavior	• The application behaves correctly for each Knob event input.

Step	Name	Description	Expected Result
5	Device Key Events	Launch an application (or test application) that can handle the Device Key events (if not already launched) Execute all supported Device key events (as observed in 3.3.2) one after another in the Client and verify the behavior	• The application behaves correctly for each input.
6	Multimedia Key events	Launch an application (or test application) that can handle the Multimedia Key events (if not already launched) Execute all supported Multimedia Key events (as observed in 3.3.2) one after another in the Client and verify the behavior	• The application behaves correctly for each input.
7	Function Key events	Launch an application (or test application) that can handle the Function Key events (if not already launched) Execute all supported Function Key events (as observed in 3.3.2) one after another in the Client and verify the behavior	• The application behaves correctly for each input.
8	ITU Keys	Skip this step if the ITU keys are not supported during the test in 3.3.2. Launch an application (or test application) that can handle the ITU Keys (if not already launched) Press and release all ITU keys one after the other. Maintain order as specified in the ITU Key Event symbol list (start with smallest value). All ITU events are pressed and released only once.	• The application behaves correctly for each input: - Key symbol values in specified ITU range - Key symbol order match the expected values - Press key events are properly closed with an appropriate release key event. - All expected key symbols are received.
9	WFD Session Teardown	See Definitions	

3.4.2 SR/WFD/USERINPUTS/PointerEvents

Requirement: MANDATORY

Condition: None

This test verifies that the MirrorLink Server correctly processes the UIBC pointer inputs received from the reference Client, which are CCC specific UIBC events under the Vendor Specific Input category as specified in [2].

This test requires a Drawing Test Application available on the MirrorLink server and supported from the MirrorLink Client.
If a Drawing Test Application is not available, Pointer Events MUST be tested through IOP test cases.

Step	Name	Description	Expected Result
1	WFD Session Setup	See Definitions If the WFD session already exists, skip this step	• The WFD session is setup and UIBC is enabled for the CCC specific UIBC capabilities between the Server and the Client • Pointer events supported during WFD capability negotiation (test case 3.3.2). The Server MUST support Pointer events and include the supported pointer event button mask in M4 Request (verify the ccc-pointer-info parameter).
2	Setup for UIBC Pointer events	Launch an application (or test application) that can handle the Pointer events (if not already launched)	• The Server is ready to receive UIBC pointer events
3	Check Pointer Events	Send a series of pointer events with w = server's framebuffer width, h = server's framebuffer height): • Press event at (w/2, h/2) • Release event at (w/2, h/2) • Press event at (w/4, h/4) • Release event at (w/2, h/4) • Continued press events from (w/2, 3h/4) to (3w/4, 3h/4) • Release at (3w/4, 3h/4) • Continued release events from (w/4, h/2) to (3w/4, h/2)	• Verify that the application behaves correctly for each pointer event input received
4	WFD Session Teardown	See Definitions	

3.4.3 SR/WFD/USERINPUTS/TouchEvents

Requirement: CONDITIONAL

Condition: Server supports touch events

This test verifies that the MirrorLink Server correctly processes the UIBC touch inputs received from the reference Client, which are CCC specific UIBC events under the Vendor Specific Input category as specified in [2].

Step	Name	Description	Expected Result
1	WFD Session Setup	See Definitions If the WFD session already exists, skip this step	• The WFD session is setup and UIBC is enabled for the CCC specific UIBC capabilities between the Server and the Client • Touch events supported during WFD capability negotiation (test case 3.3.2)

Step	Name	Description	Expected Result
2	Setup for UIBC touch events	Launch an application (or test application) that can handle the Touch events (if not already launched)	• The Server is ready to receive UIBC Touch events
3	Check Touch Events	Execute all supported touch events one by one and verify the expected behavior.	• MirrorLink Server and the application show the expected behavior on reception of each touch event from the reference Client.
4	WFD Session Teardown	See Definitions	

3.4.4 SR/WFD/USERINPUTS/Sink&SourceStatusEvents

Requirement: MANDATORY

Condition: None

This test verifies that the MirrorLink Server correctly processes the Sink Status Event message from the Client. The Server is expected to respond with a proper Source Status message. These tests are made to verify if all supported device status features such as night mode, screen saver, key lock, device lock etc. can be exchanged using UIBC Sink and Source Status events. If no Status message is sent from the Server at all, it is assumed, that the Server didn't understand the Sink Status message from the Client.

Step	Name	Description	Expected Result
1	WFD Session Setup	See Definitions If the WFD session already exists, skip this step	• The WFD session is setup and UIBC is enabled for the CCC specific UIBC capabilities between the Server and the Client
2	Setup for UIBC Sink and Source Status events	Launch an application (or test application) that can handle the Sink and Source Status events (if not already launched)	• The Server is ready to receive UIBC Sink Status events for all supported features
3	Basic Sink and Source Status events	Send a Sink Status Event from the reference Client, with all values set to unknown. Server is expected to answer with a Source Status event message.	• Valid Source Status event sent to the Client • Values can be unknown
4	Check night mode feature	Send a Sink Status event, with night mode enabled, otherwise all other values set to unknown. Server is expected to respond with the Source Status event. Test engineer is asked to verify that the night mode is active.	• Valid Source Status event arrives • Indicated night mode status is true according to the test engineer
5	Check screen saver	Send a Sink Status event, with screen saver enabled; otherwise all other values are set to unknown. Server is expected to respond with a proper Source Status message. Test engineer is asked to verify that the screen is dimmed.	• Valid Source Status event arrives • Indicated screen saver status is true according to the test engineer

Step	Name	Description	Expected Result
6	Check device-lock	Send a Sink Status event with device-lock enabled. Server is expected to respond with a proper Source Status message. Test engineer is asked to verify that the MirrorLink server device is locked.	• Valid Source Status event arrives • Indicated device-lock status is true according to the test engineer
7	Check key-lock	Send a Sink Status event with key-lock enabled. Server is expected to respond with a proper Source Status message. Test engineer is asked to verify that the keys on the MirrorLink server are disabled.	• Valid Source Status event arrives • Indicated key-lock status is true according to the test engineer
8	Check Driver Distraction Avoidance	Send a Sink Status event with driver distraction avoidance enabled. Server is expected to respond with a Source Status message. It is not possible to verify consistently that Driver Distraction Avoidance is enabled.	• Valid Source Status event arrives
9	Enable Voice Input	Send a Sink Status event with Voice Input enabled.	• Valid Source Status event arrives, with Voice Input enabled
10	Disable Voice Input	Send a Sink Status event message with Voice Input disabled.	• Valid Source Status event arrives, with Voice Input disabled
11	Check server-side FB orientation feature	Send Sink Status event to change the framebuffer orientation. The Sink Status event shall only use framebuffer orientation set to landscape. Absolute framebuffer rotation should not be used.	• Valid Source Status event arrives. Verify that Source status event only includes landscape mode in framebuffer orientation and the absolute framebuffer rotation is set to “000” only. • Orientation status is correctly set to landscape and has not changed.
12	WFD Session Teardown	See Definitions	

3.4.5 SR/WFD/USERINPUTS/UIContextEvents

Requirement: MANDATORY

Condition: None

This test verifies that the MirrorLink Server correctly sends the UI context event message to the Client.

The test engineer is asked to launch all applications available from the server device, one-by-one. The WFD Source is expected to provide the initial UI Context Event with the start of the RTP streaming. The WFD source MUST provide the context information with any new application or when such information changes. The WFD Source MUST provide UI context information for the entire display, i.e. the aggregation of the individual rectangular areas MUST always cover the entire framebuffer, and never a partial framebuffer area alone. If multiple overlapping rectangles are given, the sequence of the rectangles defines the stacking order (last rectangle on top).

Step	Name	Description	Expected Result
1	WFD Session Setup	See Definitions If the WFD session already exists, skip this step	• The WFD session is setup and UIBC is enabled for the CCC specific UIBC capabilities between the Server and the Client
2	Setup for UIBC UI Context event	Launch an application (or test application) that can handle the UI Context events (if not already launched)	• The Server is ready to send UIBC UI Context events
3	UI Context Event	Launch the next application available from the UPnP getApplicationList. Continue, until all applications have been launched	• UI Context information received at the beginning of RTP streaming • Application ID is identical to UPnP value • Trust levels are valid and identical to UPnP values • Content Category is valid and identical to UPnP value • Application Category is valid and identical to UPnP value • UI Context information contains information for the whole framebuffer
4	WFD Session Teardown	See Definitions	

3.4.6 SR/WFD/USERINPUTS/UIBlockingEvents

Requirement: MANDATORY

Condition: None

This test verifies that the MirrorLink Server correctly processes the UI blocking event message from the Client. A UI Blocking event message is sent to the Server through the UIBC channel to block the video streaming while the audio is still being needed. If the WFD Source supports the video frame skipping feature [3], it shall start it with the video frame skipping interval set to infinite.

Step	Name	Description	Expected Result
1	WFD Session Setup	See Definitions If the WFD session already exists, skip this step	• The WFD session is setup and UIBC is enabled for the CCC specific UIBC capabilities between the Server and the Client
2	Setup for UIBC UI Blocking event	Launch an application (or test application) that can handle the UI Blocking events (if not already launched)	• The Server is ready to receive UIBC UI Blocking events

Step	Name	Description	Expected Result
3	UI Blocking Event	Send UI Blocking event message with one of the following blocking reasons: • Not allowed content category • Not allowed application category • Not sufficient content trust level • Not sufficient application trust level • Content rules not followed • Not allowed application ID Re-do the tests for each blocking reason. Note: The application MAY provide the requested context information right on application launch within the first UI context information.	• If supported, the Server starts using video frame skipping feature for blocking the video, while the audio streaming still continues. If video frame skipping is not supported at the Server, the CTS discards the video frames and only renders the audio
4	Unblocking Video	Send UI Blocking event message with the blocking reason flags set to all-zeros to resume the video streaming	• The Server stops the video frame skipping feature (if active), and resume streaming video along with audio
5	WFD Session Teardown	See Definitions	

3.4.7 SR/WFD/USERINPUTS/AudioContextEvents

Requirement: MANDATORY

Condition: None

This test verifies that the MirrorLink Server correctly sends the audio context event message to the Client.

The test engineer is asked to launch all audio applications available from the server device, one-by-one. The WFD Source is expected to provide the initial audio Context Event at the beginning. The WFD source MUST provide the audio context information with any new audio application or when such information changes.

Step	Name	Description	Expected Result
1	WFD Session Setup	See Definitions If the WFD session already exists, skip this step	• The WFD session is setup and UIBC is enabled for the CCC specific UIBC capabilities between the Server and the Client
2	Setup for UIBC Audio Context event	Launch an application (or test application) that can handle the Audio Context events (if not already launched)	• The Server is ready to send UIBC Audio Context events
3	Audio Context Event	Launch the next application available from the UPnP getApplicationList. User action may be needed to initiate the audio play in the launched application. Continue, until all applications have been launched.	• Audio Context information received at the start of the application • Application ID is identical to UPnP value • Application Category is valid and identical to UPnP value

Step	Name	Description	Expected Result
4	WFD Session Teardown	See Definitions	

3.4.8 SR/WFD/USERINPUTS/AudioBlockingEvents

Requirement: MANDATORY

Condition: None

This test verifies that the MirrorLink Server correctly processes the Audio Blocking Event message from the Client. One application ID contained in the RTP stream referenced by an audio context event can be chosen and an Audio Blocking event is sent to the server in order to block this application. If the blocking is not related to any audio context event, the blocking will target any stream (0). The server is expected to react within 5 seconds, either by stopping the RTP stream or by removing the application ID from the stream.

Step	Name	Description	Expected Result
1	WFD Session Setup	See Definitions If the WFD session already exists, skip this step	• The WFD session is setup and UIBC is enabled for the CCC specific UIBC capabilities between the Server and the Client
2	RTP Server Connect	See Definitions in [10]	
3	Setup for Audio Blocking event	Launch an application (or test application) that can handle the Audio Blocking events (if not already launched) Execute the known steps to start audio streaming (if not already started with the application launch).	• RTP audio stream permitted in the Client is started • The Server is ready to receive UIBC Audio blocking events
4	Block Audio Stream	Send Audio Blocking event message. One or more application ID(s) contained in the RTP audio stream is being chosen. If no application ID was contained in the RTP audio stream, the blocking will target any stream (0). The server is expected to react within 5 seconds, by removing the RTP audio stream associated with application ID from the stream.	• Audio Context event received 200ms after the Audio Blocking event has been sent. • Application IDs do not contain the blocked application IDs.
5	Unblock Audio Stream	Send Audio Blocking event. - Application ID is the one from previous blocking notification - Blocking reason flags are all-zeros The test engineer is asked to make sure that the application is still sending audio or start the audio stream again. The server is expected to react within 5 seconds by continuing streaming audio containing those application IDs.	• Audio Context event received 200ms after the Audio Blocking event has been sent. • Application IDs contain the previously blocked application IDs.
6	RTP Server Disconnect	See Definitions in [10]	

Step	Name	Description	Expected Result
7	WFD Session Teardown	See Definitions	

3.4.9 SR/WFD/USERINPUTS/SinkCutTextEvents

Requirement: CONDITIONAL

Condition: Server supports Sink Cut Text events

This test verifies that the MirrorLink Server correctly processes the Sink Cut Text Event message from the Client. A Sink Cut Text message is sent to the Server. The Server is expected to update the framebuffer accordingly.

Step	Name	Description	Expected Result
1	WFD Session Setup	See Definitions If the WFD session already exists, skip this step	• The WFD session is setup and UIBC is enabled for the CCC specific UIBC capabilities between the Server and the Client • Sink Cut Text events supported during WFD capability negotiation (test case 3.3.2)
2	Setup for Sink Cut Text	Launch an application (or test application) that can handle the Sink Cut Text events (if not already launched)	• The Server is ready to receive UIBC Sink Cut Text events
3	Sink Cut Text	Send Sink Cut Text event with medium sized text (20-100 bytes). Send Sink Cut Text event with empty text (0 bytes). Send Sink Cut Text event with large text (>32kByte bytes).	• WFD session does not terminate. • DUT continues RTP streaming. • DUT shows received Cut Text, if supported
4	Sink Cut Unicode Text	Send Sink Cut Text event, containing the following elements, within a Single message (without the quotation marks): • "This is Latin-1 text" • 0x1B 0x25 0x67 • "This is Unicode text" • 0x03 0xA3 (Greek Σ) • 0x03 0xBC (Greek μ) • 0x00 0x1B 0x00 0x25 0x00 0x40 • "This is Latin-1 text"	• WFD session does not terminate • DUT continues RTP streaming. • DUT shows received Cut Text.
5	WFD Session Teardown	See Definitions	

3.4.10 SR/WFD/USERINPUTS/SourceCutTextEvents

Requirement: CONDITIONAL

Condition: Server supports Source Cut Text events

This test verifies that the MirrorLink Server correctly processes the Source Cut Text Event message from the Client. A Source Cut Text message is sent from the Server.

Step	Name	Description	Expected Result
1	WFD Session Setup	See Definitions If the WFD session already exists, skip this step	• The WFD session is setup and UIBC is enabled for the CCC specific UIBC capabilities between the Server and the Client • Source Cut Text events supported during WFD capability negotiation (test case 3.3.2)
2	Setup for Source Cut Text	Launch an application (or test application) that can handle the Source Cut Text events (if not already launched)	• The Server is ready to send UIBC Source Cut Text events
3	Source Cut Text	Test engineer does the necessary steps to send a Source Cut Text event.	• DUT continues RTP streaming. • Server Cut Text message received, with correct text.
4	WFD Session Teardown	See Definitions	

3.4.11 SR/WFD/USERINPUTS/TextOutputEvents

Requirement: CONDITIONAL

Condition: Server supports Text Output Events

This test verifies that the MirrorLink Server correctly sends the Text Output events to the Client. The event is sent for the correct application, and the length of the textual information should not exceed the maximum length indicated by the Client supporting this feature.

Step	Name	Description	Expected Result
1	WFD Session Setup	See Definitions If the WFD session already exists, skip this step	• The WFD session is setup and UIBC is enabled for the CCC specific UIBC capabilities between the Server and the Client • Support for the Text Output events indicated by the Client during WFD capability negotiation (test case 3.3.2)
2	Setup for Text Output events	Launch an application (or test application) that can handle the Text Output events (if not already launched)	• The Server is ready to send UIBC Text Output events
3	Text output events	The test engineer is asked to execute the known steps to send Text Output events.	• DUT continues RTP streaming. • Valid output text event arrives at the CTS with the correct application ID and the length of the returned text is smaller or equal to M.

Step	Name	Description	Expected Result
4	WFD Session Teardown	See Definitions	

4 CLIENT FEATURE TEST CASES

4.1 WFD Device Discovery at MirrorLink Client

These tests verify that the MirrorLink client is capable to perform WFD Device Discovery while operating as a WFD Sink, and sends CCC IE [3] with CCC specific OUI and the MirrorLink UPnP Device Information during the WFD device discovery phase. All other steps for verification of WFD Device Discovery are verified as mandatory tests as part of [5].

4.1.1 CL/WFD/DISCOVERY/ProbeRequestDuringScanState

Requirement: MANDATORY

Condition: None

Tools: Sniffer supporting Wi-Fi Display protocol including the required specifications it uses.

This test checks that the MirrorLink Client device advertises its capabilities including the CCC IE with CCC specific OUI and the MirorLink UPnP Device Information in the probe request frame it sends during the WFD Device Discovery at Scan state.

Step	Name	Description	Expected Result
1	Activate MirrorLink over Wi-Fi Display	Switch on WLAN radio Switch on the MirrorLink over Wi-Fi Display by using an application or user input (if not done automatically or from previous tests)	• The Client device initiates the WFD connection phase by starting the WFD discovery
2	Setting into Scan state	Put the ML Client device in Scan State with channel 6 as listen channel See Vendor Instructions for setup	• Scan covers all supported channels (both in 2.4 GHz and 5 GHz)
3	Sending Probe Requests	Start sending Probe Request frames Capture the Probe Request frames using the Sniffer Test that the CCC IE with correct information is sent in the Probe Request frames including in a non-social channel.	• CCC IE is present with OUI value set to 04 DF 69 (CCC), and the MirrorLink UPnP Device Information sub element set with the values configured for the device. The Internet Accessibility sub element may be present in the CCC IE if supported (optional). • Verify that UPnP service related information (e.g. UPnP App Server Service, Client Profile Service, Notification Server Service) are filled correctly according to the PICS for the MirrorLink Client. • The WFD device information in the WFD IE may be verified to confirm that it is set as a “Primary Sink” and the WFD Session Availability set as 0b01 “available”

Step	Name	Description	Expected Result
4	WLAN disconnection	Power off WLAN radio at the MirrorLink Server	

4.1.2 CL/WFD/DISCOVERY/ProbeRequestDuringFindState

Requirement: MANDATORY

Condition: None

Tools: Sniffer supporting Wi-Fi Display protocol including the required specifications it uses.

This test checks that the MirrorLink Client device advertises its capabilities including the CCC IE with CCC specific OUI and the MirorLink UPnP Device Information in the probe request frame it sends during the WFD Device Discovery at Find state.

Step	Name	Description	Expected Result
1	Activate MirrorLink over Wi-Fi Display	Switch on WLAN radio Switch on the MirrorLink over Wi-Fi Display by using an application or user input (if not done automatically or from previous tests)	• The Client device initiates the WFD connection phase by starting the WFD discovery
2	Setting into Find state	Put the ML Client device in device discovery phase at Find state with channel 1 as the listen channel See Vendor Instructions for setup	
3	Sending Probe Requests	Start sending Probe Request frames Capture the Probe Request frames using the Sniffer Test that the CCC IE with correct information is sent in the Probe Request frames in a social channel.	• CCC IE is present with OUI value set to 04 DF 69 (CCC), and the MirrorLink UPnP Device Information sub element set with the values configured for the device. The Internet Accessibility sub element may be present in the CCC IE if supported (optional). • Verify that UPnP service related information (e.g. UPnP App Server Service, Client Profile Service, Notification Server Service) are filled correctly according to the PICS for the MirrorLink Client. • The WFD device information in the WFD IE may be verified to confirm that it is set as a “Primary Sink” and the WFD Session Availability set as 0b01 “available”
4	WLAN disconnection	Power off WLAN radio at the MirrorLink Server	

4.1.3 CL/WFD/DISCOVERY/ProbeResponseDuringListenState

Requirement: MANDATORY

Condition: None

Testing Tools: Sniffer supporting Wi-Fi Display protocol, Packet Injector tool or a reference WFD capable MirrorLink Source device.

This test checks that the MirrorLink Client device includes its capabilities including the CCC IE with CCC specific OUI and the MirrorLink UPnP Device Information in the probe response frame it sends during the WFD Device Discovery at Listen state.

Step	Name	Description	Expected Result
1	Activate MirrorLink over Wi-Fi Display	Switch on WLAN radio Switch on the MirrorLink over Wi-Fi Display by using an application or user input (if not done automatically or from previous tests)	• The Client device initiates the WFD connection setup by starting the WFD discovery
2	Setting into Listen state	Put the ML Client device in device discovery phase at Listen state with channel 1 as listen channel See Vendor Instructions for setup	
3	Sending Test Probe Requests	Use a Packet Injector tool or a test WFD device to send several Probe Request Frames with WFD and CCC IEs, to the destination address set as broadcast address. Capture the Probe Request frames using the Sniffer	
4	Sending Probe Responses	Start sending Probe Response frames Capture the Probe Response frames using the Sniffer Test that the CCC IE with correct information is sent in the Probe Response frames including in a non-social channel.	• Probe Response from the Cline device within 1 second • CCC IE is present with OUI value set to 04 DF 69 (CCC), and the MirrorLink UPnP Device Information sub element set with the values configured for the device. The Internet Accessibility sub element may be present in the CCC IE if supported (optional). • Verify that UPnP service related information (e.g. UPnP App Server Service, Client Profile Service, Notification Server Service) are filled correctly according to the PICS for the MirrorLink Client. • The WFD device information in the WFD IE may be verified to confirm that it is set as a “Primary Sink” and the WFD Session Availability set as 0b01 “available”
5	WLAN disconnection	Power off WLAN radio at the MirrorLink Server	

4.1.4 CL/WFD/DISCOVERY/BeaconAndProbeResponseInGOMode

Requirement: MANDATORY

Condition: None

Testing Tools: Sniffer supporting Wi-Fi Display protocol, Packet Injector tool or a reference WFD capable MirrorLink Source device.

This test checks that the MirrorLink Client device includes its capabilities including the CCC IE with CCC specific OUI and the MirrorLink UPnP Device Information in the beacon and probe response frames it sends to support the WFD Device Discovery while operating as an autonomous Group Owner.

Step	Name	Description	Expected Result
1	Setting as Autonomous Group Owner	Switch on WLAN radio Put the ML Client device into the autonomous Group Owner mode with channel 11 as its operating channel and listen channel See Vendor Instructions for setup	
2	Sending Test Probe Requests	Use a Packet Injector tool or a test WFD device to send several Probe Request Frames with WFD and CCC IEs, to the destination address set as broadcast address. Capture the Probe Request frames using the Sniffer	
3	Sending Probe Responses	Start sending Probe Response frames Capture the Probe Response frames using the Sniffer Test that the CCC IE with correct information is sent in the Probe Response frames including in a non-social channel.	• Probe Response from the Client device within 1 second • CCC IE is present with OUI value set to 04 DF 69 (CCC), and the MirrorLink UPnP Device Information sub element set with the values configured for the device. The Internet Accessibility sub element may be present in the CCC IE if supported (optional). • Verify that UPnP service related information (e.g. UPnP App Server Service, Client Profile Service, Notification Server Service) are filled correctly according to the PICS for the MirrorLink Client. • The WFD device information in the WFD IE may be verified to confirm that it is set as a “Primary Sink” and the WFD Session Availability set as 0b01 “available”
4	Verification of Beacon Frames	Capture the Beacon frames transmitted by the Client device using the Sniffer	• CCC IE is present with OUI value set to 04 DF 69 (CCC), and the MirrorLink UPnP Device Information sub element set with the values configured for the device.

Step	Name	Description	Expected Result
		Test that the CCC IE with correct information is sent in the Beacon frames.	The Internet Accessibility sub element may be present in the CCC IE if supported (optional). • Verify that UPnP service related information (e.g. UPnP App Server Service, Client Profile Service, Notification Server Service) are filled correctly according to the PICS for the MirrorLink Client. • The WFD device information in the WFD IE may be verified to confirm that it is set as a “Primary Sink” and the WFD Session Availability set as 0b01 “available”
4	WLAN disconnection	Power off WLAN radio at the MirrorLink Server	

4.2 UPnP Session Setup over WFD Connection

4.2.1 CL/WFD/UPnP/UPnPSetupAfterWFDConnection

Requirement: MANDATORY

Condition: None

This test checks if the MirrorLink Client is able to start the UPnP session based on the information (CCC IE) received during the WFD Discovery (see Section 3.1) after WFD connection.

Step	Name	Description	Expected Result
1	WFD Connection Setup	See Definitions Use channel 6 as its operating channel and listen channel for 2.4 GHz Use channel 36 as operating channel if 5 GHz is supported (repeat steps 1-4)	• WFD connection is setup from the Client with P2P connection to a reference Server • The Association and 4-way handshake for authentication is captured by the Sniffer to verify the connection is completed
2	UPnP Control Point Connect	DUT accesses the DUT's Device XML. CTS starts sending SSDP:alive messages only AFTER the DUT has retrieved the Device XML.	• The DUT retrieves the Device XML from the URL provided in the CCC IE or SSDP:discover response • DUT may retrieve the Service XML. • DUT may send `SSDP:discover` message (must be formatted correctly)
3	MirrorLink Session Setup	See Definitions	
4	UPnP Control Point Disconnect	See Definitions in [7]	

4.3 WFD Session Setup

The WFD capability exchange and negotiation steps specific to MirrorLink implementation are tested under these test cases. Other tests to verify WFD capability negotiation including the support for the UIBC feature and WFD session establishment are covered by Wi-Fi Alliance certification testing as per [5].

4.3.1 CL/WFD/SESSION/SessionStartAfterApplicationLaunch

Requirement: MANDATORY

Condition: None

Note: The Client MUST support the UI blocking with video frame skipping as described in [3]. Hence, it MUST also undergo the testing for the test case "Video Frame Skipping on SoUT" in [5] as part of the Wi-Fi Alliance Miracast certification.

This test checks if the MirrorLink Client can start the WFD session setup procedure and the Audio/Video is correctly shown when the first application over WFD is launched. The WFD capability negotiation for the MirrorLink specific display resolutions and the video frame skipping feature by the Client are tested.

Step	Name	Description	Expected Result
1	WFD Connection Setup	See Definitions Use channel 1 as its operating channel and listen channel for 2.4 GHz Use channel 36 as operating channel if 5 GHz is supported (need to repeat steps 1-5)	• The WFD connection is setup with P2P link to the WFD capable reference Client device
2	UPnP Session Setup	See 4.2.1 (perform steps 2-3)	
3	WFD TCP Connection Setup	WFD TCP connection setup starts once the step 1 has been completed. This step may occur in parallel to step 2 above.	• The TCP connection is setup to the reference server, and RTSP stack is active
4	WFD Capability Negotiation	Complete RTSP M1-M4 RTSP Message Exchange as described in section 4.6 of [3]. If supported, messages may be sent manually Using the Sniffer, capture and analyze the RTSP messages during the WFD capability negotiation	• Verify that the Client supports 800x480p30 (WVGA) resolution in the M3 response by including it in the `wfd-video-formats` parameter. • Verify that if the Client indicates support for display resolution of 1280x720 or higher, the 1280x720p30 is also supported. In this case, verify that the Client includes 1280x720p30 as a supported resolution in the `wfd-video-formats` parameter in M3 response. • Verify that the Client supports video frame skipping by setting the frame-rate-control-support within wfd-video-formats parameter with b0=0b1 and b3-b1=0b000 (infinite skipping interval) in its M3 response message.

Step	Name	Description	Expected Result
5	WFD Session Establishment	Complete M5-M7 RTSP Message Exchange If supported, messages may be sent manually Using the Sniffer, capture and analyze the RTSP messages during the WFD session establishment Send a RTSP M9 request (PAUSE) to pause the streaming of audio and/or video if needed (step 2 has not completed by end of M7 request/response and Client is set in drive mode).	• Verify that the WFD session is established with RTSP OK in M7 Response sent from the reference Server
6	First application launch over WFD	Launch an application over WFD via the UPnP Launch Application. User input may be required to trigger the application. If the streaming of audio and/or video has been paused in step 7, the Client MUST send the M7 request (PLAY) to resume A/V streaming once the first application over WFD is launched.	• Verify that the Audio/video streaming starts after the M7: RTSP PLAY request/response is completed, and the Client shows the display from the launched application over WFD.
7	WFD Session Teardown	See Definitions	

4.3.2 CL/WFD/SESSION/SupportForCCCSpecificUIBC

Requirement: MANDATORY

Condition: None

This test checks if the MirrorLink Client performs the WFD capability negotiation that include correct support for CCC specific UIBC capabilities and associated parameters.

Step	Name	Description	Expected Result
1	WFD Connection Setup	See Definitions If the WFD connection already exists, this step may be skipped	• The WFD connection is setup with the WFD capable reference Server device
2	UPnP Session Setup	See 4.2.1 (perform steps 2-3)	

Step	Name	Description	Expected Result
3	First application launch over WFD	Launch an application (or test application) over WFD via the UPnP Launch Application User input may be required to trigger the application	• The TCP connection is setup to the reference Client, and RTSP stack is active
4	WFD Capability Negotiation: M1-M2 exchange	Complete RTSP M1-M2 RTSP Message Exchange If supported, messages may be sent manually	• The RTSP messages are captured by the Sniffer to verify the parameters during WFD session establishment
5	WFD Capability Negotiation: M3 request and response	Server sends RTSP GET_PARAMER request (M3) by including the `wfd_uibc_capability` parameter. Send the M3 response with the `wfd_uibc_capability` parameter indicating the supported capabilities.	• Verify that the `wfd_uibc_capability` in M3 response from the Client includes `vendor-specific-cap-info` with "04 DF 69" as the `vendor_OUI` and other required parameters as per [2], listed in Table 4-1.
6	WFD Capability Negotiation: M4 request and response	Reference Server sends RTSP SET_PARAMER (M4) request with `wfd_uibc_capability` parameter in which the ccc_event_cap_list should contain a sub-set of parameters obtained from M3 Response in Step 5. Reference Server may include the `wfd_uibc_setting` parameter set to the value 'enable' in M4 request. Send M4 response with OK from the Client.	• Verify that the Client sends the M4 response with OK after receiving the M4 request with correct parameters from the reference Server.
7	RTSP M15 request and response	Skip this test if the Server included `wfd_uibc_setting` parameter in M4 request (step 6). The Reference Server sends the RTSP SET_PARAMER request (M15) with `wfd_uibc_setting` parameter with its value set to 'enable'. Send M15 response with OK from the Client	• Verify that the Client sends the M15 response and UIBC session is established.

Step	Name	Description	Expected Result
8	WFD Session Establishment	Complete M5-M7 RTSP Message Exchange If supported, messages may be sent manually	• Verify that the WFD session is established • Verify that the Audio/video streaming starts after the M7: RTSP PLAY request/response is completed, and the Client shows the display from the launched application
9	WFD Session Teardown	See Definitions	

1

wfd-uibc-capability sub-parameter names	Format and Supported Values	Description
input-category-val	input_category_list=VENDOR_SPECIFIC	
vendor-specific-cap-info	vendor_specific_cap_info = OUI: 04 DF 69 ; ccc_event_cap_list = <ccc-event-inp-type>	
ccc-event-inp-type	ccc-resolution-info: Resolution/ [width pixel] [height pixel]	Sink display resolution
	ccc-display-info: Display/ [mm] [mm] [mm]	width x height x distance to user for Sink's display in mm
	ccc-keys-info: KnobKeys/ [4*HEXDIG value] DeviceKeys/ [4*HEXDIG value] MultimedeaKeys/ [4*HEXDIG value] FunctionKeys/ [2*HEXDIG value] ITUKeys/("none"/"Supported")	Lists the supported key events, see [2]
	ccc-pointer-info: Pointer/ [2*HEXDIG value]	Indicates the supported pointer event button mask, must be 0x00 00 if not supported
	ccc-touch-info: Touch/ [2*HEXDIG number value] [2*HEXDIG pressure mask value]	Indicates the number of supported simultaneous touch events (max number -1) and touch event pressure mask
	ccc-text-output: TextOutput/ [4*HEXDIG value]	Maximum length of textual meta information
	CutText/("none"/"Supported")	Include "Supported" if the sink supports Cut Text, otherwise "none" is included.
tcp-port	port = <IPPORT>	Port to be used for UIBC

2 Table 4-1: wfd-uibc-capability parameter values supported by CCC capable WFD Source and Sink devices

4.3.3 CL/WFD/SESSION/SessionTeardown

Requirement: MANDATORY

Condition: None

This test checks if the MirrorLink Client performs WFD session teardown when the last launched application is terminated or when the MirrorLink over WFD session ends.

Step	Name	Description	Expected Result
1	WFD Session Setup	See Definitions If the WFD session already exists, skip this step	• Audio/video streaming starts so that the Client shows the display from the launched application
2	Trigger Teardown of WFD session	CTS sends RTSP SET_PARAMETER request with trigger:TEARDOWN (M5 Request) Message to DUT	• DUT sends RTSP SET_PARAMETER response (M5 Response) Message with Status OK.
3	Teardown of WFD session	DUT is tearing down WFD session. CTS sends RTSP TEARDOWN (M8 Response) message with Status OK.	• DUT sends RTSP TEARDOWN (M8 Request) Message. • If the Sink is the P2P GO then it may send a de-authentication message to the Source and then terminate the P2P Group

4.4 WFD User Inputs and Events

4.4.1 CL/WFD/USERINPUTS/KeyEvents

Requirement: CONDITIONAL

Condition: Client supports Key Events

This test verifies that the MirrorLink Client correctly sends the UIBC key inputs, which are CCC specific UIBC events under the Vendor Specific Input category as specified in [2]. A number of key events are sent to the server. The testing verifies if the corresponding text or effect is visible on the MirrorLink Server.

This test requires a Text Test Application available on the MirrorLink Server and supported from the MirrorLink Client. If a Text Test Application is not available, X11 Key Events MUST be tested through IOP test cases.

Step	Name	Description	Expected Result
1	WFD Session Setup	See Definitions If the WFD session already exists, skip this step	• The WFD session is setup and UIBC is enabled for the CCC specific UIBC capabilities between the Server and the Client • One or more Key events supported during WFD capability negotiation (test case 4.3.2)
2	Setup for UIBC key events	Launch an application (or test application) over WFD via the UPnP Launch Application that can handle the supported key events	• The Server is ready to receive UIBC key events
3	Check X11 Key Events	Send a series of Key Press and Release events from the Client: • Letters: 'a' – 'z' • Letters: 'A' – 'Z' • Numbers: '0' – '9' • Symbols as specified: ' ', '!', '”', '#', '$', '%', '&', ''', '(', ')', '*', '+', ',', '–', '.', '/', ':', ';', '<', '=', '>', '?', '@', '[', '\', ']', '^', '_', '`', '{', '|', '}', '~', Backspace, Return	• Verify that the application behaves correctly with the key events, and the Server updates the display in accordance with the key inputs from the Client
4	Knob Key events	Launch an application (or test application) that can handle the Knob events (if not already launched) Execute all supported Knob key events (as observed in 4.3.2) one after another in the Client and verify the behavior	• The application behaves correctly for each Knob event input.

Step	Name	Description	Expected Result
5	Device Key Events	Launch an application (or test application) that can handle the Device Key events (if not already launched) Execute all supported Device key events (as observed in 4.3.2) one after another in the Client and verify the behavior	• The application behaves correctly for each input.
6	Multimedia Key events	Launch an application (or test application) that can handle the Multimedia Key events (if not already launched) Execute all supported Multimedia Key events (as observed in 4.3.2) one after another in the Client and verify the behavior	• The application behaves correctly for each input.
7	Function Key events	Launch an application (or test application) that can handle the Function Key events (if not already launched) Execute all supported Function Key events (as observed in 4.3.2) one after another in the Client and verify the behavior	• The application behaves correctly for each input.
8	ITU Keys	Skip this step if the ITU keys are not supported during the test in 4.3.2. Launch an application (or test application) that can handle the ITU Keys (if not already launched) Press and release all ITU keys one after the other. Maintain order as specified in the ITU Key Event symbol list (start with smallest value). All ITU events are pressed and released only once.	• The application behaves correctly for each input: - Key symbol values in specified ITU range - Key symbol order match the expected values - Press key events are properly closed with an appropriate release key event. - All expected key symbols are received.
9	WFD Session Teardown	See Definitions	

4.4.2 CL/WFD/USERINPUTS/PointerEvents

Requirement: CONDITIONAL

Condition: Client supports pointer events

This test verifies that the MirrorLink Client correctly sends the UIBC pointer inputs, which are CCC specific UIBC events under the Vendor Specific Input category as specified in [2]. The MirrorLink-CTS is configured

to visually track pointer events. The test engineer is asked to "draw" on the client device and watch the resulting strokes on the display. If no pointer events arrived or the test engineer did not see his drawing on the display, this test fails.

Step	Name	Description	Expected Result
1	WFD Session Setup	See Definitions If the WFD session already exists, skip this step	• The WFD session is setup and UIBC is enabled for the CCC specific UIBC capabilities between the Server and the Client • Pointer events supported during WFD capability negotiation (test case 4.3.2). CTS MUST support Pointer events and include the supported pointer event button mask in M4 Request (verify the ccc-pointer-info parameter). CTS MUST NOT enable support for Touch Events during this test.
2	Setup for UIBC Pointer events	Launch an application (or test application) that can handle the Pointer events (if not already launched)	• The CTS is ready to receive UIBC pointer events
3	Check Pointer Events	The test engineer is asked to "draw" on the client device and watch the resulting strokes on the display. If no pointer events arrived or the test engineer did not see his drawing on the display, this test fails. •	• Verify that valid Pointer Events arrive • Drawing on the client display does correspond to the input • Verify that the CTS behaves correctly for each pointer event input received
4	WFD Session Teardown	See Definitions	

4.4.3 CL/ WFD/USERINPUTS/PointerEventCoverage

Requirement: CONDITIONAL

Condition: Client supports pointer events

This test verifies whether the entire area of the framebuffer can be used for pointer interaction and whether the DUT is correctly mapping the pointer event's location into the MirrorLink Server's framebuffer dimension. The MirrorLink-CTS is configured to visually show a set of buttons. The test engineer is asked to press & release these buttons.

Step	Name	Description	Expected Result
1	WFD Session Setup	See Definitions If the WFD session already exists, skip this step	• The WFD session is setup and UIBC is enabled for the CCC specific UIBC capabilities between the Server and the Client. • Pointer events supported during WFD capability negotiation (test case 4.3.2). CTS MUST support Pointer events. CTS

Step	Name	Description	Expected Result
			MUST NOT enable support for Touch Events during this test.
2	Setup for UIBC Pointer events	Launch an application (or test application) that can handle the Pointer events (if not already launched)	• The CTS is ready to receive UIBC pointer events
3	Check Pointer Events	The CTS renders a set of buttons on the display, which the test engineer is asked to press & release (press must happen at the center of the button). Button size is following the driver distraction guidelines for button sizes. Buttons MUST be rendered at the following positions: - All 4 screen corners - Screen center	• Valid Pointer Event messages arrive (Press & Release) • Pointer Event location matches the rendered buttons • Location matches the center of the button +/- 5mm. • Verify that the CTS behaves correctly for each pointer event messages received
4	WFD Session Teardown	See Definitions	

4.4.4 CL/WFD/USERINPUTS/SimultaneousTouchEvents

Requirement: CONDITIONAL

Condition: Client supports Touch Events

This test verifies that the MirrorLink Client correctly sends the UIBC touch inputs, which are CCC specific UIBC events under the Vendor Specific Input category as specified in [2]. The MirrorLink-CTS is configured to visually track touch events. The test engineer is asked to "draw" on the client device and watch the resulting strokes on the display.

Step	Name	Description	Expected Result
1	WFD Session Setup	See Definitions If the WFD session already exists, skip this step	• The WFD session is setup and UIBC is enabled for the CCC specific UIBC capabilities between the Server and the Client • Touch events supported during WFD capability negotiation (test case 4.3.2)
2	Setup for UIBC touch events	Launch an application (or test application) that can handle the Touch events (if not already launched)	• The Server is ready to receive UIBC Touch events
3	Check Touch Events	Read number of supported simultaneous touch events from the ccc-touch-info parameter in M4 request from test case 4.3.2. Test engineer is asked to draw with as many fingers as supported from the client (not more than 10) on the MirrorLink Client's screen. The CTS tool is configured to visually track all received touch events.	• Valid Touch Events arrives • Test Engineer can visually confirm the received Touch Events. • Number of simultaneous touch events matches the number of used fingers. • All touch events are closed with a release event • Verify that the CTS behaves correctly for each touch event messages received

Step	Name	Description	Expected Result
4	WFD Session Teardown	See Definitions	

4.4.5 CL/WFD/USERINPUTS/ForceTouchEvents

Requirement: CONDITIONAL

Condition: Client supports Touch Events AND

Client supports more than 2 pressure levels

This test verifies that the MirrorLink Client correctly sends the UIBC touch inputs, which are CCC specific UIBC events under the Vendor Specific Input category as specified in [2]. The MirrorLink-CTS is configured to visually track touch events. The test engineer is asked to "draw" on the client device and watch the resulting strokes on the display.

Step	Name	Description	Expected Result
1	WFD Session Setup	See Definitions If the WFD session already exists, skip this step	• The WFD session is setup and UIBC is enabled for the CCC specific UIBC capabilities between the Server and the Client • Touch events supported during WFD capability negotiation (test case 4.3.2)
2	Setup for UIBC touch events	Launch an application (or test application) that can handle the Touch events (if not already launched)	• The Server is ready to receive UIBC Touch events
3	Check Touch Events	Read number of supported pressure events from the ccc-touch-info parameter in M4 request from test case 4.3.2. Test engineer is asked to press with one finger with different pressure forces on the MirrorLink Client's touch screen. The CTS tool is configured to visually show the pressure level of the received touch events.	• Valid Touch Event messages arrives • Test Engineer can visually confirm different pressure levels. • All touch events are closed with a release event • Verify that the CTS behaves correctly for each touch event messages received
4	WFD Session Teardown	See Definitions	

4.4.6 CL/WFD/USERINPUTS/Sink&SourceStatusEvents

Requirement: MANDATORY

Condition: None

This test verifies that the MirrorLink Client correctly sends the Sink Status Event message. The Server is expected to respond with a proper Source Status message. The tests are made to verify if all supported device status features such as night mode, screen saver, key lock, device lock etc. can be exchanged using UIBC Sink and Source Status events. If no Status message is sent from the Server at all, it is assumed, that the Server didn't understand the Sink Status message from the Client.

Step	Name	Description	Expected Result
1	WFD Session Setup	See Definitions If the WFD session already exists, skip this step	• The WFD session is setup and UIBC is enabled for the CCC specific UIBC capabilities between the Server and the Client
2	Setup for UIBC Sink Status event	Launch an application (or test application) that can handle the Sink and Source Status events (if not already launched	• The Server is ready to receive UIBC Sink Status events for all supported features
3	Check night mode feature	Test engineer changes day/night switch to Night on DUT	Skip test case, if DUT does not support Night Mode switch.
		Wait for Sink Status event message. Send Source Status event message	• Valid Source Status event received • Night Mode flag is set to '11'
		Test engineer changes a client day/night switch to Day	
		Wait for Sink Status event message. Send Source Status event message	• Valid Source Status event received • Night Mode flag is set to '10'
4	Check key-lock	Test engineer changes a client Key Lock switch to ON	Skip test case, if DUT does not support Key Lock switch.
		Wait for Sink Status event message. Send Source Status event message	• Valid Source Status event received • Key Lock flag is set to '11'
		Test engineer changes a client Key Lock switch to OFF	
		Wait for Sink Status event message. Send Source Status event message	• Valid Source Status event received • Key Lock flag is set to '10'
5	Check Driver Distraction Avoidance	Test engineer changes a client Driver Distraction Avoidance switch to ON (enable Distraction Avoidance)	
		Wait for Sink Status event message. Send Source Status event message.	• Valid Source Status event received • Driver Distraction Avoidance flag is set to '11'
		Test engineer changes a client Driver Distraction Avoidance switch to OFF (disable Distraction Avoidance)	
		Wait for Sink Status event message. Send Source Status event message	• Valid Source Status event received • Driver Distraction Avoidance flag is set to '10'
6	Check device-lock	Test engineer changes a client Device Lock switch to ON	Skip test case, if DUT does not support Device Lock switch.
		Wait for Sink Status event message. Send Source Status event message	• Valid Source Status event received • Device Lock flag is set to '11'
		Test engineer changes a client Device Lock switch to OFF	

Step	Name	Description	Expected Result
		Wait for Sink Status event message. Send Source Status event message	• Valid Source Status event received • Device Lock flag is set to '10'
7	Check Voice Input	Test engineer changes a client Voice Input switch to ON (enable voice input)	Skip test case, if DUT does not support Voice Input switch.
		Wait for Sink Status event message. Send Source Status event message	• Valid Source Status event received • Voice Input flag is set to '11'
		Test engineer changes a client Voice Input switch to OFF (disable voice input)	
		Wait for Sink Status event message. Send Source Status event message	• Valid Source Status event received • Voice Input flag is set to '10'
8	Check Microphone Input	Send Source Status event with Microphone Input set to '11' (to open Microphone)	Skip test case, if DUT does not support Microphone Input. • Audio connection established, if not already done
		Test Engineer is asked to speak into the MirrorLink Client's microphone	• Receive audio over an established audio connection
		Send Source Status event with Microphone Input set to '10' (to close Microphone)	• Audio connection may be disconnected
		Test Engineer is asked to speak into the MirrorLink Client's microphone	• No audio received
9	WFD Session Teardown	See Definitions	

4.4.7 CL/WFD/USERINPUTS/UIBlockingEvents

Requirement: MANDATORY

Condition: None

This test verifies that the MirrorLink Client correctly sends the UI blocking event message to the Server.

The test case expects the MirrorLink Client to display one application (with allowed context information) and then to block another application (with not-allowed context information). The Client sends UI Blocking event message to the Server through the UIBC channel to block the video streaming while the audio is still being needed. The Client MUST support the UI blocking with video frame skipping as described in [3], with the video frame skipping interval set to infinite.

Step	Name	Description	Expected Result
1	WFD Session Setup	See Definitions If the WFD session already exists, skip this step	• The WFD session is setup and UIBC is enabled for the CCC specific UIBC capabilities between the Server and the Client
2	Setup for UIBC UI Blocking event	Launch an application (or test application) with allowed context information (if not already launched)	• The Server is ready to receive UIBC UI Blocking events

Step	Name	Description	Expected Result
3	UI Blocking Event	Execute the known step to start an application with not-allowed context information that will not be allowed from MirrorLink Client	• The Client sends correct UI Blocking event • Application IDs match the UI context information rectangles • Blocking reason includes the application ID
5	WFD Session Teardown	See Definitions	

4.4.8 CL/WFD/USERINPUTS/AudioBlockingEvents

Requirement: MANDATORY

Condition: None

This test verifies that the MirrorLink Client correctly sends the Audio Blocking Event message to the Server. The test case expects the MirrorLink Client to play audio from one application (with allowed context information) and to block audio from another application (with not-allowed context information).

Step	Name	Description	Expected Result
1	WFD Session Setup	See Definitions If the WFD session already exists, skip this step	• The WFD session is setup and UIBC is enabled for the CCC specific UIBC capabilities between the Server and the Client
2	RTP Client Connect	See Definitions in [10]	
3	Setup for Audio Blocking event	Launch an application (or test application) that can handle the Audio Blocking events (if not already launched) Execute the known step to start audio streaming, which is allowed from MirrorLink Client (if not already started with the application launch).	• RTP audio stream permitted in the Client is started • The Server is ready to receive UIBC Audio blocking events
4	Audio Blocking	RTP server sends an audio stream with not-allowed context information Send Audio Blocking event message. One or more application ID(s) contained in the RTP stream is being chosen.	• Receive at least one Audio Blocking event from the Client through UIBC • Blocking reason includes the correct application ID.
5	RTP Client Disconnect	See Definitions in [10]	
6	WFD Session Teardown	See Definitions	

4.4.9 CL/WFD/USERINPUTS/SinkCutTextEvent

Requirement: CONDITIONAL

Condition: Client supports Sink Cut Text events

This test verifies that the MirrorLink Client correctly processes the Sink Cut Text Event message from the Server.

Step	Name	Description	Expected Result
1	WFD Session Setup	See Definitions If the WFD session already exists, skip this step	• The WFD session is setup and UIBC is enabled for the CCC specific UIBC capabilities between the Server and the Client • Sink Cut Text events supported during WFD capability negotiation (test case 4.3.2)
2	Setup for Sink Cut Text	Launch an application (or test application) that can handle the Sink Cut Text events (if not already launched)	• The Server is ready to send UIBC Sink Cut Text events
3	Sink Cut Text	Test engineer does the necessary steps to send a Sink Cut Text event.	• DUT continues to display the RTP stream. • Sink Cut Text message received, with correct text.
4	WFD Session Teardown	See Definitions	

4.4.10 CL/WFD/USERINPUTS/SourceCutTextEvent

Requirement: CONDITIONAL

Condition: Client supports Source Cut Text events

This test verifies that the MirrorLink Client correctly processes the Source Cut Text Event message from the Server.

Step	Name	Description	Expected Result
1	WFD Session Setup	See Definitions If the WFD session already exists, skip this step	• The WFD session is setup and UIBC is enabled for the CCC specific UIBC capabilities between the Server and the Client • Source Cut Text events supported during WFD capability negotiation (test case 4.3.2)
2	Setup for Source Cut Text	Launch an application (or test application) that can handle the Source Cut Text events (if not already launched)	• The Server is ready to receive UIBC Source Cut Text events
3	Source Cut Text	Send Source Cut Text event with medium sized text (20-100 bytes). Send Source Cut Text event with empty text (0 bytes). Send Source Cut Text event with large text (>32kByte bytes).	• WFD session does not terminate. • UI visible on DUT • DUT shows received Cut Text, if supported

Step	Name	Description	Expected Result
4	Source Cut Unicode Text	Send Source Cut Text event, containing the following elements, within a Single message (without the quotation marks): • "This is Latin-1 text" • 0x1B 0x25 0x67 • "This is Unicode text" • 0x03 0xA3 (Greek Σ) • 0x03 0xBC (Greek μ) • 0x00 0x1B 0x00 0x25 0x00 0x40 • "This is Latin-1 text"	• WFD session does not terminate • UI visible on DUT • DUT shows received Cut Text, if supported.
5	WFD Session Teardown	See Definitions	

4.4.11 CL/WFD/USERINPUTS/TextOutputEvents

Requirement: CONDITIONAL

Condition: Client supports Text Output Events

This test verifies that the MirrorLink Client correctly receives the Text Output events from the Server.

Step	Name	Description	Expected Result
1	WFD Session Setup	See Definitions If the WFD session already exists, skip this step	• The WFD session is setup and UIBC is enabled for the CCC specific UIBC capabilities between the Server and the Client • Text Output events supported during WFD capability negotiation (test case 4.3.2)
2	Setup for Text Output events	Launch an application (or test application) that can handle the Text Output events (if not already launched) Using the verification results during test case 4.3.2, record the maximum length of output text supported by the Client (say M)	• The Client is ready to receive UIBC Text Output events
3	Text output events	CTS sends a Text Output event with - Maximum length < M - Maximum length = M - Maximum length > M	• WFD session does not terminate • UI visible on DUT • DUT shows received Output Text (text longer than M is truncated or ignored).
4	WFD Session Teardown	See Definitions	

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 – MirrorLink over Wi-Fi Display", Version 1.2.0 CCC-TS-049

[3] Wi-Fi Display Technical Specification Version 1.0.0, September 2012

[4] Car Connectivity Consortium, "MirrorLink – IEEE 802.11 CCC Information Element", Version 1.2, CCC-TS-050

[5] Wi-Fi CERTIFIED Miracast™ Interoperability Test Plan, Version 1.0.0, August 2012

[6] Wi-Fi Peer To Peer (P2P) Technical specification, v1.2

[7] CCC-TS-031-MirrorLink_UPnP Server Device Test Specification, v1.1.1

[8] Car Connectivity Consortium, "MirrorLink – core Architecture", Version 1.2, CCC-TS-032

[9] Car Connectivity Consortium, "MirrorLink – VNC Based Display and Control", Version 1.1, CCC-TS-010.

[10] Car Connectivity Consortium, "MirrorLink – Audio Test Specification", Version 1.1, CCC-TS-013

[11] Car Connectivity Consortium, "Device Certification Program Management Document (PMD)", CCC-PR-002.