



**Publicly Available Specification (PAS);
Intelligent Transport Systems (ITS);
MirrorLink[®];
Part 17: MirrorLink over Wi-Fi Display (WFD)**

CAUTION

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

CCC is owner of the copyright of the document CC-TS-049 and/or had all relevant rights and had assigned said rights to ETSI on an "as is basis". Consequently, to the fullest extent permitted by law, ETSI disclaims all warranties whether express, implied, statutory or otherwise including but not limited to merchantability, non-infringement of any intellectual property rights of third parties. No warranty is given about the accuracy and the completeness of the content of the present document.

Reference

DTS/ITS-88-17

Keywords

interface, ITS, PAS, smartphone

ETSI

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

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

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

Important notice

The present document can be downloaded from:

<http://www.etsi.org/standards-search>

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the only prevailing document is the print of the Portable Document Format (PDF) version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status.

Information on the current status of this and other ETSI documents is available at

<https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx>

If you find errors in the present document, please send your comment to one of the following services:

<https://portal.etsi.org/People/CommiteeSupportStaff.aspx>

Copyright Notification

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.

The content of the PDF version shall not be modified without the written authorization of ETSI.

The copyright and the foregoing restriction extend to reproduction in all media.

©ETSI 2017.

© Car Connectivity Consortium 2011-2017.

All rights reserved.

ETSI logo is a Trade Mark of ETSI registered for the benefit of its Members.

MirrorLink® is a registered trademark of Car Connectivity Consortium LLC.

RFB® and VNC® are registered trademarks of RealVNC Ltd.

UPnP® is a registered trademark of UPnP Forum.

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

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members.

3GPP™ and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

oneM2M logo is protected for the benefit of its Members.

GSM® and the GSM logo are trademarks registered and owned by the GSM Association.

Contents

Intellectual Property Rights	4
Foreword.....	4
Modal verbs terminology.....	4
1 Scope	5
2 References	5
2.1 Normative references	5
2.2 Informative references.....	6
3 Definitions and abbreviations.....	6
3.1 Definitions.....	6
3.2 Abbreviations	6
4 Introduction	6
5 MirrorLink over WFD Procedure.....	8
5.1 General	8
5.2 Phase 1: WFD Connection Setup	9
5.3 Phase 2: UPnP Setup	10
5.4 Phase 3: WFD Session Setup	11
5.5 Phase 4: WFD Operation.....	13
5.6 Terminating a MirrorLink over WFD Session	15
6 WFD Audio.....	15
6.1 Audio Links.....	15
6.2 WFD Audio Forward Channel	16
6.3 WFD Audio Back Channel.....	16
6.4 Telephony over WFD.....	16
7 WFD User Input	16
7.1 General	16
7.2 UIBC Input Body Format for MirrorLink	17
7.2.1 General.....	17
7.2.2 Key Event	18
7.2.3 Pointer Event	18
7.2.4 Touch Event.....	18
7.2.5 Sink & Source Status Events	19
7.2.6 UI Context Event	19
7.2.7 UI Blocking Event	20
7.2.8 Audio Context Event	20
7.2.9 Audio Blocking Event	21
7.2.10 Sink& Source Cut Text Events	22
7.3 MirrorLink Specific RTSP Data Structures.....	22
7.3.1 General.....	22
7.3.2 wfd-uibc-capability parameter	22
7.3.3 wfd-uibc-setting parameter	24
8 WFD Content Protection.....	24
Annex A (normative): Key Event Table.....	25
Annex B (informative): Authors and Contributors.....	27
History	28

Intellectual Property Rights

Essential patents

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

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

Trademarks

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

Foreword

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

The present document is part 17 of a multi-part deliverable. Full details of the entire series can be found in part 1 [i.1].

Modal verbs terminology

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

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

1 Scope

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

The present specification describes the integration of Wi-Fi Display to MirrorLink.

2 References

2.1 Normative references

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

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

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

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

- [1] ETSI TS 103 544-2 (V1.3.0): "Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink®; Part 2: Virtual Network Computing (VNC) based Display and Control".
- [2] ETSI TS 103 544-3 (V1.3.0): "Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink®; Part 3: Audio" .
- [3] ETSI TS 103 544-18 (V1.3.0): "Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink® ; Part 18: IEEE 802.11TM Car Connectivity Consortium (CCC) Information Element".
- [4] ETSI TS 103 544-12 (V1.3.0): "Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink®; Part 12: UPnP Server Device".
- [5] Wi-Fi Alliance® Technical Committee, Wi-Fi Display Technical Task Group: "Wi-Fi Display Technical Specification" Version 1.0.0, September 2012, <https://www.scribd.com/doc/250439511/Wi-Fi-Display-Technical-Specification-v1-0-0>.
- [6] ETSI TS 103 544-9 (V1.3.0): "Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink®; Part 9: UPnP Application Server Service".
- [7] ETSI TS 103 544-13 (V1.3.0): "Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink®; Part 13: Core Architecture".
- [8] IETF RFC 6143: "The Remote Framebuffer Protocol", March 2011, <https://tools.ietf.org/html/rfc6143>.
- [9] Void.
- [10] IETF RFC 2131: "Dynamic Host Configuration Protocol", March 1997, <https://tools.ietf.org/html/rfc2131>.
- [11] IETF RFC 2132: "DHCP Options and BOOTP Vendor Extensions", March 1997, <https://tools.ietf.org/html/rfc2132>.

2.2 Informative references

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

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

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

- [i.1] ETSI TS 103 544-1 (V1.3.0): "Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink®; Part 1: Connectivity".
- [i.2] Wi-Fi Alliance®: "Best Practices Document for Wi-Fi CERTIFIED Miracast™ Devices", Version 1.0, September 2014.

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following definitions apply:

sink device: device that receives multimedia content from a WFD source over a Wi-Fi link and renders it

source device: device that supports streaming multimedia content to a WFD sink(s) over a Wi-Fi link

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ABC	Audio Back Channel
HDCP	High-bandwidth Digital Content Protection
IE	Information Element
IEEE	Institute of Electrical and Electronics Engineers
Miracast	Commercial denomination of WFD
ML	MirrorLink
OUI	Organizationally Unique Identifier
PES	Packetized Elementary Stream
UIBC	User Input Back Channel
UPnP	Universal Plug and Play
USB	Universal Serial Bus
VNC	Virtual Network Computing
WFD	Wi-Fi Display which is the technology and specification being officially called "Wi-Fi Alliance Wi-Fi Display specification"

4 Introduction

Wi-Fi Display, also known as Miracast, is a peer-to-peer wireless screen replication standard created by the Wi-Fi Alliance. Its main purpose is to let the source device project its screen to the sink device screen, and to provide the sink device with the method to control the source device.

Figure 1 shows the typical Client/Server topology for the MirrorLink over Wi-Fi Display.

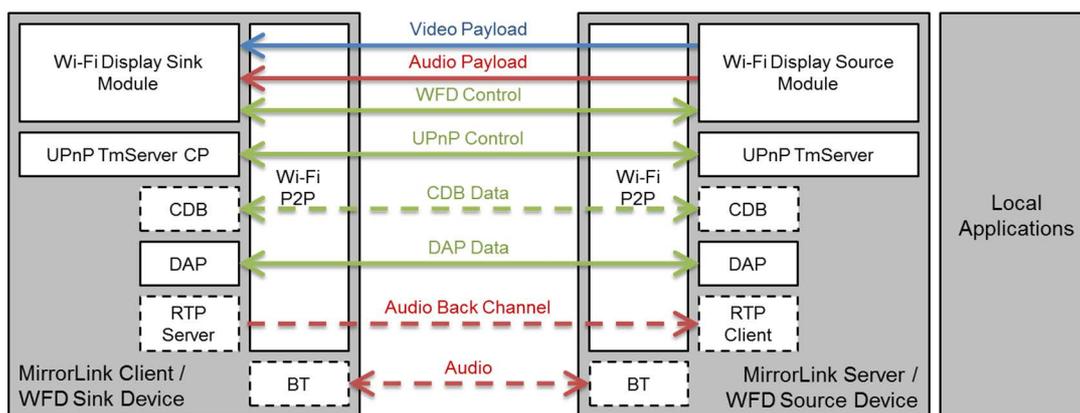


Figure 1: High Level Topology

The present document specifies the integration of Wi-Fi Display into MirrorLink, providing an alternative video link to VNC. The specification of the other MirrorLink components, like UPNP, CDB, DAP etc. is done in their respective documents.

The MirrorLink Client, providing WFD functionality, shall implement the WFD Sink functionality.

The MirrorLink Server, providing WFD functionality, shall implement the WFD Source functionality.

NOTE: The term "Sink" used in the present document refers to a WFD Primary Sink device as defined in [5].

Figure 2 displays the layered architecture diagram for the integration of WFD into MirrorLink. WFD stack is added to MirrorLink stack. The diagram applies to both Client and Server devices, which shall apply it according to their roles.

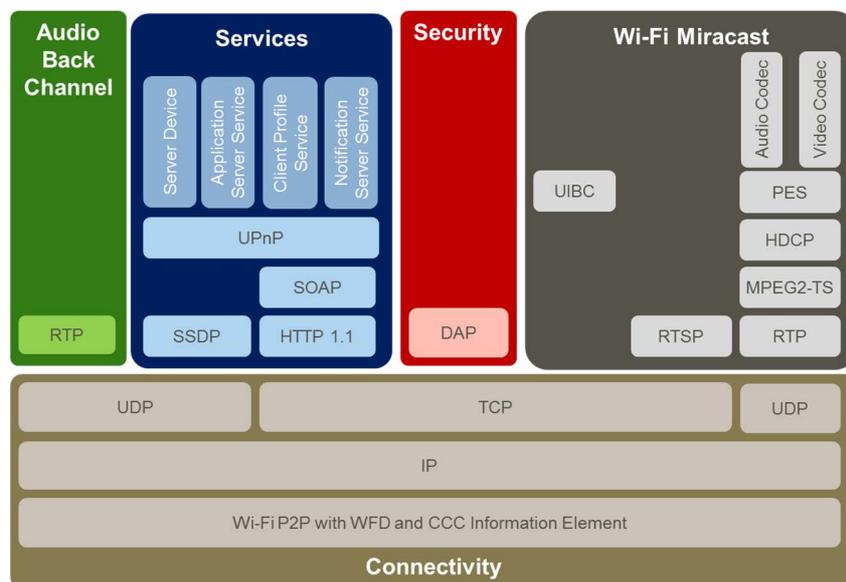


Figure 2: MirrorLink over Wi-Fi Display Architecture

Through Wi-Fi Display, MirrorLink Server and Client discover each other with the Wi-Fi Device Discovery procedure, which exchange the Information Element. WFD UIBC is integrated into MirrorLink stack for User Input.

MirrorLink Client and Server shall support all Wi-Fi Display mandatory functions and services, as described in [5], Table 3-1. This includes the following functions and services:

- WFD Device Discovery with IE for CCC
- WFD Connection Setup
- WFD Capability negotiation

- WFD Session establishment
- Encoding and packetization of the captured Display
- Transport of multiplexed audio and video payload
- De-multiplex, de-packetization and decode of received audio and video payload
- Rendering of decoded video on local display panel
- Power Save mechanisms
- Session termination
- Encode and packetization of captured audio
- Multiplex video and audio payload
- Rendering of decoded audio on local speakers
- AV Stream Control using RTSP

The MirrorLink Client and Server shall support the following optional Wi-Fi Display functions:

- User Input Back Channel (UIBC)

Use of BT HFP in accordance with the MirrorLink Audio Specification [2] shall be possible for the MirrorLink over WFD implementation as well.

5 MirrorLink over WFD Procedure

5.1 General

MirrorLink over Wi-Fi Display (WFD) connection between MirrorLink Server acting as WFD source device and MirrorLink Client acting as WFD sink device shall take place in the 4 following phases, as depicted in Figure 3.

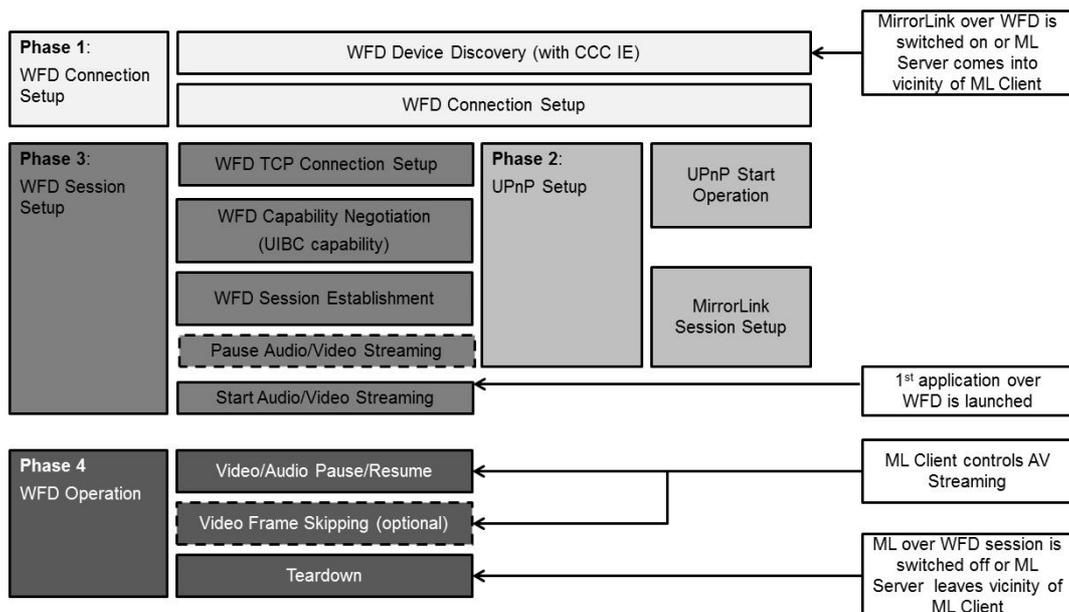


Figure 3: MirrorLink over Wi-Fi Display Diagram

5.2 Phase 1: WFD Connection Setup

Phase 1 shall start when MirrorLink over Wi-Fi Display is switched on. In addition, if persistent WFD Group for MirrorLink exists, it is recommended that WFD Connection setup proceeds automatically without user interaction such as re-selection of WFD & CCC capable device.

The following requirements apply to the phase 1:

WFD Device Discovery:

To establish a MirrorLink over Wi-Fi Display connection, Wi-Fi P2P device discovery with WFD IE (Information Element) shall be used. Wi-Fi Display devices shall advertise the WFD IEs defined in Wi-Fi Display specification.

In addition to the WFD IEs, the MirrorLink devices shall include the CCC Information Element that shall contain the MirrorLink UPnP Device Information sub-element and may contain the Internet Accessibility sub-element, as specified in [3], into all beacon, probe request and probe response frames. The MirrorLink devices shall detect other MirrorLink devices through CCC IE.

Detection of the CCC Information Element, together with the WFD Information Element, is sufficient for a Wi-Fi P2P device to determine that the sending Wi-Fi P2P device is supporting MirrorLink over WFD. In case Tunneled Data Link Service (TDLS) is used, the CCC Information shall also be tunneled through the Wi-Fi Access Point.

In case a MirrorLink Server is a Wi-Fi P2P concurrent device, i.e. connected to the MirrorLink Client's Access Point as a WLAN STA and implements a Wi-Fi P2P device, both interfaces will belong to different Basic Service Sets (BSS). In this case the MirrorLink Client and Server should use regular Wi-Fi P2P connection setup to establish the WFD session.

Wi-Fi P2P defines 2 phases for device discovery (refer to [5] for details). In the *Scan* phase, P2P devices collect information about surrounding devices or networks by scanning all supported channels. Devices may actively send Probe Request frames to look for Legacy AP or P2P Group Owners. The *Find* phase has two states, a *Search* and a *Listen* state. Within the *Search* state, P2P devices go through a fixed set of channels and send Probe Requests. Within the *Listen* state, P2P devices wait on a fixed channel and respond to received Probe Requests.

In case of an existing P2P Group, an existing P2P GO, or an existing Legacy AP (using TDLS), successful discovery is possible in *Scan* phase, assuming that all devices are operational. In case no P2P Group exists, one device has to be in *Find/Search* state, while the other device has to be in *Find/Listen* state to ensure successful device discovery.

Typically, most MirrorLink Servers will only be in *Scan* phase in regular intervals. Additionally, a MirrorLink Server may only go into *Scan* phase after a user interaction. Therefore, establishment of a MirrorLink over WFD connection may need manual interaction from the consumer, at least for the initial connection setup.

The MirrorLink Client, supporting MirrorLink over WFD, should switch on Wi-Fi P2P and start the WFD Device Discovery, latest when the MirrorLink Client gets powered on, otherwise the MirrorLink Client shall provide a consumer accessible mechanism, which will enable WFD Device Discovery.

The MirrorLink Server, supporting MirrorLink over WFD, shall provide a consumer accessible mechanism, which will enable the Wi-Fi P2P Device Discovery. This mechanism may only be usable, after the consumer has switched on the Wi-Fi radio.

The mechanism to enable WFD Device Discovery will only ensure the Wi-Fi P2P connection setup. This should not automatically start the WFD stream.

WFD Connection Setup

A WFD connection setup using Wi-Fi P2P shall be supported. A WFD connection setup using TDLS should not be used.

The MirrorLink devices shall follow the process of Wi-Fi P2P/WFD as specified in [5].

The MirrorLink devices shall connect to a device, which includes a WFD IE and a CCC IE. To establish a P2P connection for a WFD connection setup, the MirrorLink devices shall also include the CCC Information

Element that shall contain the MirrorLink UPnP Device Information sub-element and may contain the Internet Accessibility sub-element, as specified in [3], when transmitting the P2P Invitation Request, P2P Invitation Response, GO Negotiation Request, GO Negotiation Response, GO Confirmation, Association/Reassociation Request and Association/Reassociation Response frames.

The Persistence WFD Group allows automatic WFD connection through caching the information for the Group. To establish a Persistence WFD Group, the MirrorLink devices should follow the process of WFD as specified in [5].

WFD Automatic Re-Connection

The MirrorLink Server and Client should allow for automatic reconnection, in case the devices are known, and known to have used MirrorLink over WFD recently.

In case automatic reconnection is supported, the MirrorLink Client shall either automatically switch on Wi-Fi P2P and start the WFD Device Discovery or shall automatically switch on Bluetooth. In case it recognizes a known MirrorLink Server to which it had previously connected via MirrorLink over WFD, it shall automatically attempt to reconnect, unless automatic reconnection is disabled from the consumer.

In case automatic reconnection is supported, the MirrorLink Server shall either automatically switch on Wi-Fi P2P and start the WFD Device Discover or shall automatically switch on Bluetooth. In case it recognizes a known MirrorLink Client to which it had previously connected via MirrorLink over WFD, it shall automatically attempt to reconnect, unless automatic reconnection is disabled from the consumer.

MirrorLink Clients and Servers supporting automatic reconnection should implement Persistent WFD Group over P2P.

Testing Considerations:

MirrorLink devices may refuse to establish a Wi-Fi P2P connection to devices not capable of supporting MirrorLink. In order to undergo Miracast conformance and IOP testing, those devices may implement a specific test mode in which a Wi-Fi P2P session with non-MirrorLink devices is possible, which may be disabled afterwards for MirrorLink operation.

5.3 Phase 2: UPnP Setup

Phase 1 shall be completed, before phase 2 can start.

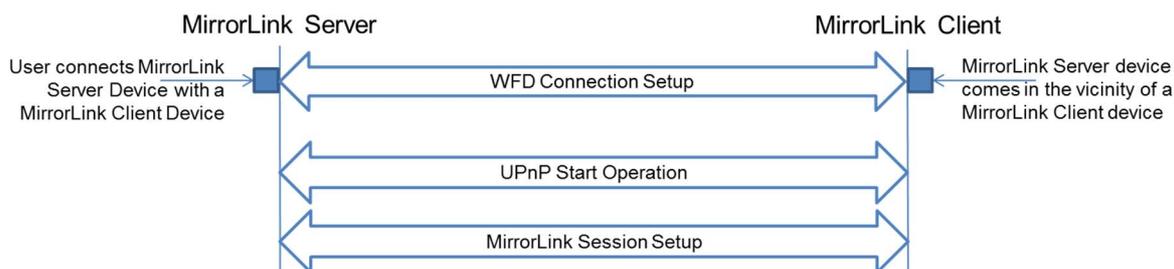


Figure 4: UPnP Setup Sequence Diagram

UPnP Operation Start

Based on the information within the CCC IE from the WFD device discovery and WFD connection setup stages, the MirrorLink device activates the respective UPnP components (i.e. *TmServer* Control Point or *TmServerServer*) in the device.

The MirrorLink Client shall immediately activate its UPnP *TmServerDevice:1* Control Point if it connects to a MirrorLink device with a UPnP *TmServerDevice:1* Server.

The MirrorLink Server shall immediately activate its UPnP *TmServerDevice:1* Server, if it connects to a MirrorLink device with a UPnP *TmServerDevice:1* Control Point.

The MirrorLink Client shall follow the UPnP Operation Start sequence defined in the MirrorLink Core Architecture specification [7].

UPnP Device Description

The MirrorLink Client devices shall either use the CCC Information Element, defined in [3], or shall send an `SSDP:discover` message to determine the location of the MirrorLink Server's UPnP Server Device XML. The MirrorLink Client shall not wait for an `SSDP:alive` messages for initial UPnP Setup.

The structure of the UPnP Server Device XML is specified in [4]. The UPnP Server Device XML description shall be accessible at the following URL via HTTP-GET.

```
http://<IPAddress>:<Port Number>/<path>
```

The MirrorLink Client shall form the UPnP Device Description URL with the following elements, when using the CCC Information Element:

- **IP Address:** IP address of the MirrorLink Server, as retrieved with in the DHCP negotiation with the WFD connection setup process. The `DHCPOFFER` message includes both, the DHCP Server and Client IP address. The IP address of the DHCP Server can be determined during the DHCP negotiation, according to IETF RFC 2131 [10]. A DHCP server always returns its own address in the '*server identifier*' option, as defined in IETF RFC 2132 [11].
- **Port Number:** Port number of the MirrorLink UPnP Server, as provided in the MirrorLink UPnP Device Information sub-elements of the CCC Information Element as specified in [3].
- **Path:** Static path "`TmServerDevice/TmServerDevice:1.xml`".

EXAMPLE:

```
http://192.168.3.15:2869/TmServerDevice/TmServerDevice:1.xml
```

The MirrorLink Server shall provide the same URL to its Device XML via the CCC Information Element and in response to the `SSDP:discover` message. Latest after the MirrorLink Client has accessed the Device XML, the MirrorLink Server shall start the `SSDP:alive` advertisements. In case the MirrorLink Server goes temporarily offline, it shall send an `SSDP:byebye` message followed by a `SSDP:alive` message, when becoming online again.

UPnP Service Description

The MirrorLink Client shall follow the Core Specification [7].

MirrorLink Session Setup:

The MirrorLink Client shall follow the MirrorLink Session Setup sequence defined in the MirrorLink Core Architecture specification [7].

The MirrorLink Client shall set its Client Profile using UPnP *SetClientProfile* action over the established Wi-Fi connection.

Testing Considerations:

In order to undergo Miracast conformance and IOP testing, MirrorLink devices may implement a specific test mode in which the UPnP Setup is bypassed. The test mode may include bypassing the setup of DAP, CDB, and RTP connections.

5.4 Phase 3: WFD Session Setup

Phase 3 may start in parallel to the Phase 2, unless the WFD session has been setup outside the MirrorLink session. WFD TCP connection shall be established for the purpose of WFD RTSP procedures within the time limit specified in [5]. Similarly, the WFD session setup shall start after the establishment of TCP connection to ensure that the WFD RTSP timeout requirements in [5] are met.

After successful exchange of RTSP M7 request/response, as shown in Figure 5, the WFD Sink may immediately send a RTSP M9 request (PAUSE) to pause the streaming of audio and/or video content from the WFD Source to the WFD Sink. The application launch is typically triggered from the user. The WFD Sink shall send the M7 request (PLAY) to resume A/V streaming once the 1st application over WFD is launched if AV streaming has been previously paused. Phase 3 shall not be executed, if the WFD session is already setup, e.g. triggered from a previous UPnP Application Launch action.

Testing Considerations:

MirrorLink devices may refuse to establish a WFD session to devices not capable of supporting MirrorLink over WFD. In order to undergo Miracast conformance and IOP testing, those devices may implement a specific test mode in which a WFD session with non-MirrorLink WFD devices is possible, which may be disabled afterwards for MirrorLink operation.

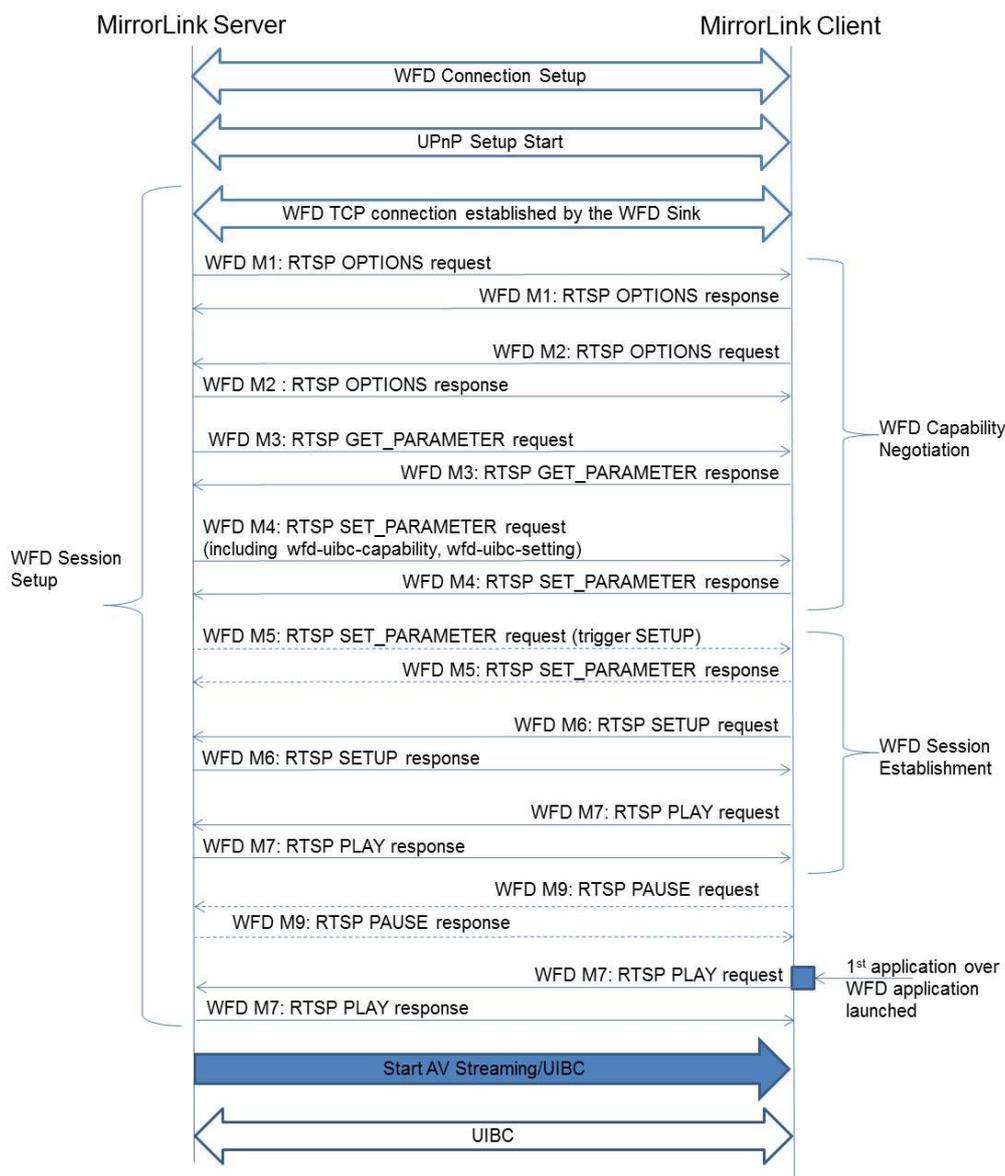


Figure 5: WFD Session Setup Procedure

The following is assumed before A/V streaming in a WFD Session can start:

- WFD connection has been established between the MirrorLink Server and MirrorLink Client.
- UPnP Session has been established.
- MirrorLink Server device has been attested.

- WFD session has been setup.
- UIBC has been established and enabled.

The A/V streaming in WFD session shall be started only after the launch of the first application via UPnP, which has a protocol identifier (protocolID) value of "WFD".

NOTE: There is no specific "WFD application", which is advertised separately via UPnP and which has to be launched prior being able to use any user application over WFD.

The WFD Session Setup shall follow the process of WFD Display Specification as described in [5], and consist of a number of M1, M2, M3, M4, M5, M6 and M7 messages exchanges.

NOTE: For further information of the messages flow of WFD Session Setup, clauses 4.6 and 4.8 in [5] can give detail process and description.

NOTE: For further information of the WFD messages used in messages flow, clause 6 in [5] can give detail information of format and description.

For the specific UIBC for the car usage, M3/M4 messages shall contain *wfd-uibc-capability* parameter. The *wfd-uibc-setting* parameter shall be included in the first RTSP M4 request message during the WFD Capability Negotiation.

WFD Source shall not set any capability in the RTSP M4 Request which are not indicated to be supported by the WFD Sink in its RTSP M3 Response.

Once the WFD session has been successfully established with the UIBC enabled, the MirrorLink Server or MirrorLink Client shall not send any M15 Request (RTSP SET_PARAMETER) message to disable the UIBC.

The MirrorLink Server starts Audio/Video streaming to the MirrorLink Client after receiving the RTSP PLAY message from the MirrorLink Client.

The MirrorLink Server and Client shall support 800x480p30 as the baseline configuration. Additionally, both WFD Sink and Source should support a resolution, which is close to their native framebuffer resolution. The MirrorLink Server should select a resolution, which is closest to the MirrorLink Client's native resolution, as provided in its M3 message (*ccc-resolution-info*). The WFD Source shall select a resolution, which is equal or exceeding the MirrorLink Reference Screen resolution (800x480p30). The WFD Sink shall scale the received WFD framebuffer to the WFD Sink device's native resolution.

Applications shall render using the highest resolution supported by the MirrorLink Client and Server as determined during the WFD capability negotiation. They shall preserve the aspect ratio of the negotiated resolution, while not clipping. The MirrorLink Server shall add padding if required. The MirrorLink Server shall not stretch its framebuffer to compensate for any difference in the framebuffer aspect ratio.

The ML Server shall not send the display content if the application does not support landscape and only launches in Portrait orientation when the ML Client is in drive mode. When the ML Client is in park mode, the ML Server may transmit the framebuffer in Landscape even if the application launches in Portrait orientation.

5.5 Phase 4: WFD Operation

Phase 2 and 3 shall be completed, before phase 4 can start.

During the WFD Operation phase, the WFD session is controlled from the MirrorLink Client, using RTSP commands and UIBC events. This allows the MirrorLink Client to start or pause the streaming of the MirrorLink Server's framebuffer. In case the MirrorLink Client does not need the WFD session anymore, it can tear-down the session.

The MirrorLink Client is capable of controlling the MirrorLink Server Audio/Video streaming.

Pause Video, while Audio Continues

The sequence to pause the RTP video streaming, while audio is still being played is shown in figure 6.

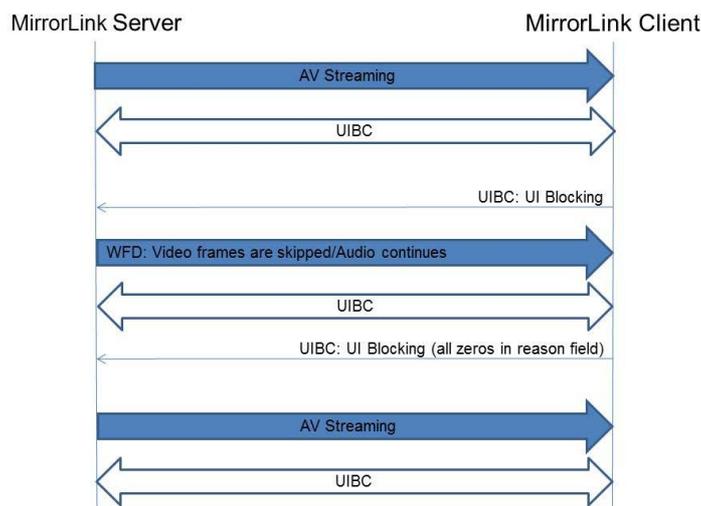


Figure 6: WFD Operation – Video-only Pause Sequence Diagram

In this case, WFD capability needs to negotiate for video frame skipping to be supported with video frame skipping interval set to infinite. Refer to Wi-Fi Display Specification [5], clause 4.10.3.1 for details about video frame skipping feature in WFD.

The MirrorLink Client shall send a UI blocking message over the WFD UIBC channel with the reason flag "UI not visible on remote display" enabled, when it intends to pause the RTP video streaming, while audio is still being needed.

The MirrorLink Client shall support the video frame skipping feature, specified in Wi-Fi Display Specification [5], clause 4.10.3.1. The MirrorLink Client shall set the *Max Skip Interval* bits (B3:B1) in the *frame-rate-control-bitmap* included in the *wfd-video-formats* parameter in the RTSP M3 response message to all zeros, indicating that the maximum allowable time interval is unspecified.

The MirrorLink Server may use the video frame skipping feature to start skipping video frames. Audio will continue to be streamed.

The MirrorLink Client shall send a UI blocking message with all reason flags set to zeros over WFD UIBC channel, if it intends to resume the RTP video streaming. The MirrorLink Server shall stop using the video frame skipping feature and continue to stream audio and video at the negotiated rates, if the MirrorLink Server had enabled framebuffer skipping.

Pause Audio/Video

The sequence to pause the RTP Audio/Video streaming is shown in Figure 7.

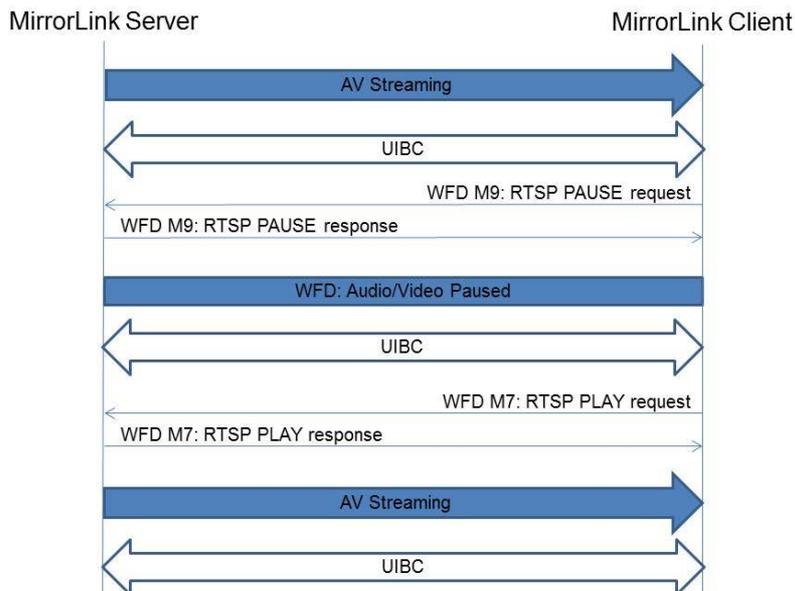


Figure 7: WFD Operation – Audio/Video Pause Sequence Diagram

The MirrorLink Client shall send a WFD M9 message (RTSP PAUSE) if it intends to pause the RTP audio/video stream. The MirrorLink Server shall acknowledge the RTSP PAUSE message and stop AV streaming.

The MirrorLink Client shall send a WFD M7 message (RTSP PLAY), if it intends to resume the RTP audio/video stream. The MirrorLink Server shall acknowledge the RTSP PLAY message and resume AV streaming.

Teardown

The MirrorLink Client shall not issue an RTSP TEARDOWN command, unless the MirrorLink session ends or the MirrorLink Client is switching to another remote UI mechanism.

Otherwise, the MirrorLink Server/Client can reconfigure the WFD session within the limits of the WFD specification.

5.6 Terminating a MirrorLink over WFD Session

MirrorLink Server and Client shall follow the procedures defined in the Core Specification [7] to disconnect a MirrorLink over WFD session. To avoid that graceful disconnection is leading to long response timeout, if the WFD peer is already disconnected or not responding anymore, the following behavior should be implemented:

A terminating MirrorLink Client shall send a Set Client Profile message with an empty Client Profile, as defined, but may disconnect prior receiving the UPnP response.

A terminating MirrorLink Server shall send SSDP : :byebye messages.

In any case, immediate loss of Wi-Fi P2P connectivity shall be considered a termination of the ML Session, and shall not lead to an unresponsive behavior.

6 WFD Audio

6.1 Audio Links

The selection of Audio links is specified in the respective Audio Specification [2].

6.2 WFD Audio Forward Channel

A MirrorLink Server shall use WFD's AV transport to provide media audio to the MirrorLink Client. The MirrorLink Client shall not launch the MirrorLink Server's RTP Server.

BT A2DP shall not be used together with WFD. An existing BT A2DP connection to the MirrorLink Client shall be disconnected by the MirrorLink Client. After a WFD session termination, the MirrorLink Client and Server should not automatically reconnect the BT A2DP connection.

6.3 WFD Audio Back Channel

WFD does not provide an Audio Back Channel.

In case the MirrorLink Client intends to setup an Audio Back Channel, it shall follow the regular MirrorLink mechanism, as specified in the Audio Specification [2]. In case the MirrorLink Client intends to use RTP, it shall launch the MirrorLink Server's RTP Client. In case the MirrorLink Client intends to use BT BVRA, it shall launch the MirrorLink Server's BT HFP endpoint.

The Audio Back Channel should be established before the first WFD Application is launched from the MirrorLink Client.

6.4 Telephony over WFD

BT HFP should be used to implement telephony functionality over WFD. In this case, the MirrorLink Client shall launch the MirrorLink Server's BT HFP endpoint. The MirrorLink Client may use RTP based telephony audio, if supported from the MirrorLink Server. In this case, the MirrorLink Client shall launch the MirrorLink Server's RTP Client.

7 WFD User Input

7.1 General

MirrorLink Server and MirrorLink Client shall support User Input Back Channel as a mandatory feature of MirrorLink, through which the user input from user interface displayed at MirrorLink Client could be communicated back to MirrorLink Server. In addition, user interface related output and status events are exchanged through the User Input Back Channel.

The User Input Back Channel will also be used for certain messages in the forward direction (MirrorLink Server/WFD Source to MirrorLink Client/WFD Sink). Traditionally, the UIBC channel is used for messages going from the WFD Sink to the WFD Source but in MirrorLink, this channel will be used for messages going in both directions.

Messages from MirrorLink Client to MirrorLink Server include the following types:

- Key event;
- Pointer event;
- Touch event;
- Sink Status event;
- UI Blocking event;
- Audio Blocking event;
- Sink Cut Text event.

Messages from MirrorLink Server to MirrorLink Client include the following types:

- Source Status event;
- UI Context event;
- Audio Context event;
- Source Cut Text event.

All UIBC user input & output shall be supported by using an Input Category set to a value 15 (defined in [5] as a Reserved input category, the Generic and HIDC categories may not be supported in MirrorLink setup). The Input Category value 15 is used as a Vendor-Specific category for MirrorLink. The Vendor Specific input body format for MirrorLink is described in clause 7.2.

7.2 UIBC Input Body Format for MirrorLink

7.2.1 General

The payload structure for packetizing UIBC user inputs are specified in clause 4.11.1 (UIBC Data Encapsulation) of [5].

NOTE: The UIBC input body field should be padded up to an integer multiple of 16 bits to have an even integer number in the Length field as recommended in the WFD specification [5].

All MirrorLink specific UIBC inputs shall use the Input Category field set to Vendor-Specific input category (value set to 15), and the format for the UIBC Input Body field is as shown in Table 1 below:

Table 1: UIBC Input Message Format for MirrorLink

Field	Size (Octet)	Value
OUI	3	04-DF-69 (CCC specific OUI)
Type ID	1	User Input type event ID as listed in Table 2
Length	2	Length of the following fields in octets
Descriptor	Variable	The details of the user inputs

The specified input type events are listed below in Table 2 together with their obligation for the MirrorLink Server and Client. The support and capabilities of some input events are available from the provided field within the *wfd-uibc-capability* parameter (specified later in clause 7.3.2). If the parameter field is provided as "None", the input event does not have a corresponding field in the *wfd-uibc-capability* parameter.

Table 2: Type IDs for Vendor-Specific UIBC Category for MirrorLink

Type ID	Notes	Origin	wfd-uibc-capability parameter field	Obligation
0	Key Event	Sink	ccc-keys-info	Shall for ML Server Should for ML Client
1	Pointer Event	Sink	ccc-pointer-info	Shall for ML Server Should for ML Client
2	Touch Event	Sink	ccc-touch-info	Should
3	Sink Status Event	Sink	None	Shall
4	Source Status Event	Source	None	Shall
5	UI Context Event	Source	None	Shall
6	UI Blocking Event	Sink	None	Shall
7	Audio Context Event	Source	None	Shall
8	Audio Blocking Event	Sink	None	Shall
9	Sink Cut Text Event	Sink	ccc-inp-type	Should
10	Source Cut Text Event	Source	ccc-inp-type	Should

Type ID	Notes	Origin	wfd-uibc-capability parameter field	Obligation
11	Text Output Event	Source	ccc-text-output	Deprecated
12-255	Reserved			

7.2.2 Key Event

The User Input Descriptor field of the MirrorLink UIBC Input message for the Key Event Input Type ID is shown in Table 3.

Table 3: User Input Descriptor Field for Key Events

Field	Size (Octet)	Notes
Down Flag	1	Non-zero (true) if the key is now pressed, zero (false) if it is now released.
Key Symbol Value	4	Key Symbol Value

The Key Symbol Values shall be as specified in the RFB specification [8] and in annex A. Key Events shall be handled similar to the VNC behavior expressed in [1]. A MirrorLink Client, not supporting Pointer events, shall support the knob key events listed in the Event Configuration Messages section of [1].

7.2.3 Pointer Event

The User Input Descriptor field of the MirrorLink UIBC Input message for the Pointer Event Input Type ID is shown in Table 4. The pointer event indicates either pointer movement or a pointer button press or release.

Table 4: User Input Descriptor Field for Pointer Events

Field	Size (Octet)	Notes
Button Mask	1	Current states of buttons 1 to 8 are represented by bits 0 to 7 of the <i>button-mask</i> respectively. Each bit value set to 0 means up and 1 means down (pressed).
X-position	2	X-coordinate of the pointer
Y-position	2	Y-coordinate for the pointer

A single touch events may be send as a touch event input with one single touch event ($N = 0x01$) as defined in clause 7.2.4, if supported from the MirrorLink Client and Server. Touch and pointer events should not be mixed.

The MirrorLink Client shall provide the coordinates within the framebuffer resolution of the current WFD session.

7.2.4 Touch Event

User Input Descriptor field of the MirrorLink UIBC Input message for the Touch Event Input Type ID is shown in Table 5. Touch events are used to describe touch screen action in which the user touches the screen with multiple individual fingers at different locations. The coordinate origin (0, 0) is defined to be the top-left corner of the rectangular display region.

Table 5: User Input Descriptor Field for Touch Events

Field	Size (Octet)	Notes
Number of events (N)	1	Number of individual events of a multi-location touch event. When set to 0x01, it indicates a single touch event.
For $i = 1: N$ {	$N \times 6$	
Individual touch event	6	Individual event, consisting of a (x,y) coordinate, an event identifier and a pressure value.

Field	Size (Octet)	Notes
		The format of the individual event is specified in the Touch Event clause of [1]
}		

The WFD sink shall only use event identifier within the range $[0; N_{max}-1]$, where N_{max} is the maximum number of simultaneous supported touch events, as exchanged within the UIBC ccc-touch-num-info parameter as described in clause 7.3.2. Each event shall be completed, i.e. each press event shall be later followed by a release event.

NOTE: The UIBC header has a time stamp (2-byte field), which should be used as a time reference for gesture recognition.

The MirrorLink Client shall provide the coordinates within the framebuffer resolution of the current WFD session.

7.2.5 Sink & Source Status Events

The User Input Descriptor field of the MirrorLink UIBC Input message for the Sink & Source Status Event Type ID is shown in Table 6. The Sink & Source Status event provides status information of specific device features and the ability to set them.

Table 6: User Input Descriptor Field for Sink & Source Status Events

Field	Size (Octet)	Notes
Status	4	Status of device features as specified in the Device Status Messages clause of [1].

7.2.6 UI Context Event

The User Input Descriptor field of the MirrorLink UIBC Input message for the UI Context Event Type ID is shown in Table 7.

Table 7: User Input Descriptor Field for UI Context Events

Field	Size (Octet)	Notes
Reference	4	Context reference within the RTP UI stream. This Reference field shall be set to a non-zero value; the value 0 is reserved.
Number (N)	1	Number of UI context information.
For i = 1: N {	N x 24	
X	2	X-position of rectangle (top left corner)
Y	2	Y-position of rectangle (top left corner)
Width	2	Width of rectangle
Height	2	Height of rectangle
App ID	4	Unique application identifier, the UI is originating from. Applications being advertised via UPnP, shall match the advertised AppID; otherwise set to zero.
Trust level	2	Trust Level for Application category (see [6], Table 6-1).
Trust level	2	Trust Level for Content category (see [6], Table 6-1).
Application category	4	Application category (see [6], Table 6-2).
Content category	4	Content category (see [6], Table 6-3).
}		

The WFD Source shall provide the initial UI Context Events with the start of the RTP streaming. The WFD Sink should not show any content, prior the initial UI Context Event has been received. The WFD Source shall provide context information, whenever there is a change in the context information,

If the WFD Sink receives a UI Context event, with an application category set to "Switch to MirrorLink Client native UI" (0xF000FFFF) the MirrorLink Client shall switch to a native user-interface. Additionally, MirrorLink Client shall send a UI blocking event, defined in clause 7.2.7, with *Reason* set to "UI not visible on remote display". This allows the MirrorLink Server to enable the video frame skipping feature, if supported from the MirrorLink Client, as defined in clause 5.5. The MirrorLink Client should not pause the WFD stream, in response, to allow audio to continue in the background.

The WFD Source shall provide context information for the entire display, i.e. the aggregation of the individual rectangular areas shall 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).

7.2.7 UI Blocking Event

The User Input Descriptor field of the MirrorLink UIBC Input message for the UI Blocking Event Type ID is shown in Table 8.

Table 8: User Input Descriptor Field for UI Blocking Events

Field	Size (Octet)	Notes
Reference	4	Context reference to the RTP UI stream. The UI blocking reference shall be equal to the UI context event's reference, which is subject to block. Shall be zero if blocking is not related to any UI context event.
Number (M)	1	Number of UI blocking information.
For i = 1: M {		
Index	1	UI context information index (N) starting at one, which is subject to block. The Index shall be zero if blocking is not related to any UI context information.
Reason	2	Reason for blocking UI (bitmask) The bit mask is specified in the "Reason for Blocking" entry in the Framebuffer Blocking Notification Message section of [1]. All-zero reason flags in this field indicates that the referenced RTP UI stream shall be unblocked.
}		

The WFD Sink may send an UI Blocking event, with the reason flags being all-zeros, to indicate that the RTP UI stream shall be unblocked.

The Reference field is used to uniquely identify a UI Context event sent from the Source to the Sink. In order to block the framebuffer of an application, the WFD Sink should send a UI Blocking event using the same Reference field and UI context information index.

Details on how the MirrorLink Server and Client shall handle the blocking of the User Interface are defined in the VNC specification, Framebuffer Blocking Notification section of [1].

7.2.8 Audio Context Event

The User Input Descriptor field of the MirrorLink UIBC Input message for the Audio Context Event Type ID is shown in Table 9.

Table 9: User Input Descriptor Field for Audio Context Events

Field	Size (Octet)	Notes
Reference	4	Context reference to the RTP audio stream. This Reference field shall be set to a non-zero value; the value 0 is reserved.
Number (N)	1	Number of audio context information. The audio context information number shall be set to zero (0), if the RTP stream does not carry real audio anymore.
For i = 1: N {	N x 8	Note: No list, if N equals zero (0).
App ID	4	Unique application identifier, the audio is originating from. Applications being advertised via UPnP, shall match the advertised AppID; otherwise set to zero.
Application category	4	Application category (see [6], Table 6-2).
}		

Setting the audio context information number to zero, is meant as an indication from the MirrorLink Server (and its applications), that intentionally no further audio is going to be provided at the moment. The MirrorLink Server may resume the audio playback at any later time though.

The MirrorLink Server shall send an Audio Context Event, when the value of Number (N) changes.

The MirrorLink Client should use the Number of audio context information as a trigger to:

- Fade In/Out to local audio sources, if N becomes zero and audio is available from the MirrorLink Client.
- Fade In/Out to MirrorLink audio sources, if N becomes non-zero and no other higher-priority audio is available from the MirrorLink Client.

7.2.9 Audio Blocking Event

The User Input Descriptor field of the MirrorLink UIBC Input message for the Audio Blocking Event Type ID is shown in Table 10.

Table 10: User Input Descriptor Field for Audio Blocking Events

Field	Size (Octet)	Notes
Reference	4	Context reference within the RTP audio stream. The audio blocking reference shall be equal to the audio context event's reference, which is subject to block. Shall be zero if blocking is not related to any audio context event.
Number (M)	1	Number of audio blocking information.
For i = 1: M {		
Index	1	Audio context information index (N) starting at one, which is subject to block. The Index shall be zero if blocking is not related to any audio context information.
Reason	2	Reason for blocking audio (bitmask) The bitmask is specified in the "Reason for blocking" entry in the Audio Blocking Notification Message section of [1].
}		

The Reference field is used to uniquely identify an audio Context event sent from the Source to the Sink. In order to block the audio of an application, the WFD Sink shall send an audio blocking event using the same Reference field and the Audio context information index.

Details on how the MirrorLink Server and Client shall handle the blocking of the Audio is defined in the Audio Blocking Notification section of [1].

7.2.10 Sink& Source Cut Text Events

The User Input Descriptor field of the MirrorLink UIBC Input message for the Sink & Source Cut Text Event Input Type ID is shown in Table 11. Ends of lines are represented by the linefeed / newline character (value 0x000A) alone. No carriage-return (value 0x000D) is needed.

Table 11: User Input Descriptor Field for Sink & Source Cut Text Events

Field	Size (Octet)	Notes
Length	4	Number of UTF16 characters of the text content
Text	<i>Length</i>	Text content as an array of UTF16 characters of the specified length

7.3 MirrorLink Specific RTSP Data Structures

7.3.1 General

The UIBC capability negotiation and update process are achieved using RTSP procedures.

MirrorLink Client and Servers shall use UIBC (User Input Back Channel) to implement the User Input & Output Event mechanisms. During RTSP capability negotiation phase, *wfd-uibc-capability* parameter is used to describe what UIBC related attributes are supported, and the *wfd-uibc-setting* parameter is used to enable the UIBC session. The MirrorLink specific event inputs are included as the *vendor-specific-cap-info* parameter within the *wfd-uibc-capability*.

7.3.2 wfd-uibc-capability parameter

When using WFD for MirrorLink, the *wfd-uibc-capability* parameter (as described in clause 6.1.15 of [5]) is used with *input-cat* field set to "VENDOR_SPECIFIC" and the *vendor-specific-cap-info* field set as specified (in the bold-face) as below.

```

wfd-uibc-capability           = "wfd_uibc_capability:" SP ("none" / (input-category-
                                val ";" generic-cap-val ";" hidc-cap-val ";" vendor-
                                specific-cap-info ";" tcp-port)) CRLF; "none" if not
                                supported
input-category-val            = "input_category_list=" ("none" / input-category-list)
input-category-list           = input-cat * ("," SP input-category-list)
input-cat                     = "GENERIC" / "HIDC" / "VENDOR_SPECIFIC"
generic-cap-val               = "generic_cap_list=" ("none" / generic-cap-list)
generic-cap-list              = inp-type * ("," SP generic-cap-list)
inp-type                      = "Keyboard" / "Mouse" / "SingleTouch" / "MultiTouch" /
                                "Joystick" / "Camera" / "Gesture" / "RemoteControl"
hidc-cap-val                  = "hidc_cap_list=" ("none" / hidc-cap-list)
hidc-cap-list                 = detailed-cap * ("," SP hidc-cap-list)
detailed-cap                  = inp-type "/" inp-path
inp-path                      = "Infrared" / "USB" / "BT" / "Zigbee" / "Wi-Fi" / "No-
                                SP"
vendor-specific-cap-info      = "vendor_specific_cap_info=" ("none" / ("OUI:" SP
                                vendor-OUI ";" vendor-specific-cap-list))
vendor-OUI                    = 6*6HEXDIG; value set to "04DF69" for CCC OUI when
                                used in MirrorLink setup
vendor-specific-cap-list      = ccc-event-cap-val; list to include one or more vendor
                                specific UIBC related capability parameters
ccc-event-cap-val             = "ccc_event_cap_list=" ("none" / ccc-event-inp-type)
ccc-event-inp-type            = ccc-resolution-info SP ccc-display-info SP ccc-keys-
                                info SP ccc-pointer-info SP ccc-touch-info SP ccc-
                                text-output SP ccc-cut-text

```

ccc-resolution-info	= "Resolution/" ("none" / (ccc-resolution-width-info SP ccc-resolution-height-info))
ccc-resolution-width-info	= 4*4HEXDIG; Sink display resolution width [pixel]
ccc-resolution-height-info	= 4*4HEXDIG; Sink display resolution height [pixel]
ccc-display-info	= "Display/" ("none" / (ccc-display-width-info SP ccc-display-height-info SP ccc-display-distance-info))
ccc-display-width-info	= 4*4HEXDIG; Sink display width [mm]
ccc-display-height-info	= 4*4HEXDIG; Sink display height [mm]
ccc-display-distance-info	= 4*4HEXDIG; Sink display distance to user [mm]
ccc-keys-info	= ccc-keys-knob-info SP ccc-keys-device-info SP ccc-keys-multimedia-info SP ccc-keys-function-info SP ccc-keys-itu-info
ccc-keys-knob-info	= "KnobKeys/" 8*8HEXDIG; knob keys (Bit mask according to Knob Shift and Rotate Configuration Settings table of [1])
ccc-keys-device-info	= "DeviceKeys/" 8*8HEXDIG; device keys (Bit mask according to Key Event Mapping table of [1])
ccc-keys-multimedia-info	= "MultimediaKeys/" 8*8HEXDIG; multimedia keys (Bit mask according to Key Event Mapping table of [1])
ccc-keys-function-info	= "FunctionKeys/" 2*2HEXDIG; number of additional function keys
ccc-keys-itu-info	= "ITUKeys/none"; deprecated
ccc-pointer-info	= "Pointer/" 2*2HEXDIG; pointer event button mask (according to [1])
ccc-touch-info	= "Touch/" ccc-touch-num-info SP ccc-touch-pressure-info
ccc-touch-num-info	= 2*2HEXDIG; number of supported simultaneous touch events -1
ccc-touch-pressure-info	= 2*2HEXDIG; touch event pressure mask
ccc-text-output	= "TextOutput/0x0000" 4*4HEXDIG; deprecated
ccc-cut-text	= "CutText/" ("none" / "Supported"); none if not supported
tcp-port	= "port=" ("none" / IPPORT)

The *ccc-resolution-info* and *ccc-display-info* fields shall be supported from WFD Sinks.

All other fields of *wfd-uibc-capability* parameters are set as specified in [5].

When the MirrorLink Client does not support some of the keys or events included in the *ccc-event-cap-val* field of the *wfd-uibc-capability* parameter, the field values that are not supported shall be set as per the table 12.

Table 12: Setting of ccc-event-cap-val field in the wfd-uibc-capability parameter

ccc-event-cap-val sub-parameters in wfd-uibc-capability	Fields included by the ML Client	Field value to be set if not Supported by the ML Client
ccc-event-inp-type	ccc-resolution-info	N/A (it shall be set to non-zero pixel values)
	ccc-display-info	N/A (it shall be set to non-zero values for width and height. Distance may be set to 0 if unknown)
	ccc-keys-info	Setting for sub-parameters included below
	ccc-pointer-info	Pointer/0x00
	ccc-touch-info	Touch/0x00 0x00
	ccc-text-output	TextOutput/0x0000
	ccc-cut-text	CutText/none
ccc-keys-info:	ccc-keys-knob-info	KnobKeys/0x00000000
	ccc-keys-device-info	DeviceKeys/0x00000000
	ccc-keys-multimedia-info	MultimediaKeys/0x00000000
	ccc-keys-function-info	FunctionKeys/0x00

ccc-event-cap-val sub-parameters in wfd-uibc-capability	Fields included by the ML Client	Field value to be set if not Supported by the ML Client
	ccc-keys-itu-info	ITUKeys/none

Capability setting of the above events shall be identical to the capability setting for the VNC session as defined in the VNC specification [1].

7.3.3 wfd-uibc-setting parameter

When using WFD for MirrorLink, the *wfd-uibc-setting* parameter (as described in clause 6.1.16 of [5]) SHALL be used during WFD capability negotiation or with the M15 RTSP message with *uibc-setting* field set to "enable" to start the UIBC session.

8 WFD Content Protection

Content protection using HDCP may be used by following the procedures described in [5]. Additional technical guidance and recommended best practices for MirrorLink devices implementing WFD with HDCP is provided in [i.2]. When using ML over WFD, the ML server shall not require HDCP content protection to start streaming of A/V content not requiring protection.

The WFD Source shall inject a message into the video stream, showing an Error message to the user, if no HDCP session can be established, as described in clause 3.3.3 of [i.2].

Annex A (normative): Key Event Table

The Key event mapping for different 2D knobs is shown in table A.1. The key event mapping for a particular head-unit knob *n* shall be according the following format:

0 x 3 0 0 0 0 0 n m

The value *n* defines the head-unit knob and *m* defines the event as defined in the template above. Allowed values for *n* are [0:3] and for *m* are [0:F].

Table A.1

Category	Mnemonic	KeySymValue	Description
Knob Keys	Knob_2D_n_shift_right	0x3000 00n0	Right shift
	Knob_2D_n_shift_left	0x3000 00n1	Left shift
	Knob_2D_n_shift_up	0x3000 00n2	Up shift
	Knob_2D_n_shift_up_right	0x3000 00n3	Up & right shift
	Knob_2D_n_shift_up_left	0x3000 00n4	Up & left shift
	Knob_2D_n_shift_down	0x3000 00n5	Down shift
	Knob_2D_n_shift_down_right	0x3000 00n6	Down & right shift
	Knob_2D_n_shift_down_left	0x3000 00n7	Down & left shift
	Knob_2D_n_shift_push	0x3000 00n8	Push
	Knob_2D_n_shift_pull	0x3000 00n9	Pull
	Knob_2D_n_rotate_x	0x3000 00nA	x clockwise rotation
	Knob_2D_n_rotate_X	0x3000 00nB	x anti-clockwise rotation
	Knob_2D_n_rotate_y	0x3000 00nC	y clockwise rotation
	Knob_2D_n_rotate_Y	0x3000 00nD	y anti-clockwise rotation
	Knob_2D_n_rotate_z	0x3000 00nE	z clockwise rotation
Knob_2D_n_rotate_Z	0x3000 00nF	z anti-clockwise rotation	
ITU Keys	ITU_Key_0	0x3000 0100	0, ' '
	ITU_Key_1	0x3000 0101	1, '!', ''
	ITU_Key_2	0x3000 0102	2, a, b, c
	ITU_Key_3	0x3000 0103	3, d, e, f
	ITU_Key_4	0x3000 0104	4, g, h, i
	ITU_Key_5	0x3000 0105	5, j, k, l
	ITU_Key_6	0x3000 0106	6, m, n, 0
	ITU_Key_7	0x3000 0107	7, p,q, r, s
	ITU_Key_8	0x3000 0108	8, t, u, v
	ITU_Key_9	0x3000 0109	9, w, x, y, z
	ITU_Key_Asterix	0x3000 010A	*, +
ITU_Key_Pound	0x3000 010B	#, shift	
Device Keys	Device_Phone_call	0x3000 0200	Take a phone call
	Device_Phone_end	0x3000 0201	End phone call
	Device_Soft_left	0x3000 0202	Left soft key
	Device_Soft_middle	0x3000 0203	Middle soft key
	Device_Soft_right	0x3000 0204	Right soft key

Category	Mnemonic	KeySymValue	Description
	Device_Application	0x3000 0205	Shortcut to the Application listing
	Device_Ok	0x3000 0206	Ok
	Device_Delete	0x3000 0207	Delete (Backspace)
	Device_Zoom_in	0x3000 0208	Zoom in
	Device_Zoom_out	0x3000 0209	Zoom out
	Device_Clear	0x3000 020A	Clear
	Device_Forward	0x3000 020B	Go one step forward
	Device_Backward	0x3000 020C	Go one step backward
	Device_Home	0x3000 020D	Shortcut to the Home Screen
	Device_Search	0x3000 020E	Shortcut to the search function
	Device_Menu	0x3000 020F	Shortcut to the (application) menu
	Function Keys	Function_Key_0	0x3000 0300
Function_Key_1		0x3000 0301	
...		...	
Function_Key_254		0x3000 03FE	Reserved
Function_Key_255		0x3000 03FF	
Multimedia Keys	Multimedia_Play	0x3000 0400	Start media playing
	Multimedia_Pause	0x3000 0401	Pause media playing
	Multimedia_Stop	0x3000 0402	Stop media playing
	Multimedia_Forward	0x3000 0403	Forward
	Multimedia_Rewind	0x3000 0404	Rewind
	Multimedia_Next	0x3000 0405	Go to next track in playlist
	Multimedia_Previous	0x3000 0406	Go to previous track in playlist
	Multimedia_Mute	0x3000 0407	Mute the audio stream at source
	Multimedia_Unmute	0x3000 0408	Unmute the audio stream at source
	Multimedia_Photo	0x3000 0409	Take a photo

Annex B (informative): Authors and Contributors

The following people have contributed to the present document:

Rapporteur:	Dr. Jörg Brakensiek, E-Qualus (for Car Connectivity Consortium LLC)
Other contributors:	Fan Bai, General Motors
	Matthias Benesch, Mercedes-Benz Research & Development North America
	Juan V. Jativa-Villoldo, Nokia
	Sungjin Park, Samsung Electronics
	Padam Kafle, Qualcomm
	Fawad Shaukat, Qualcomm
	Vijay Subramaniam, Qualcomm
	Viji Raveendran, Qualcomm

History

Document history		
V1.3.0	October 2017	Publication