# Car Connectivity Consortium MirrorLink®

# **Common API**

Version 1.1.7 & 1.2.2 (CCC-TS-038)



Copyright © 2011-2015 Car Connectivity Consortium LLC

All rights reserved

Confidential

# **VERSION HISTORY**

Version	Date	Comment
1.1.0	20 December 2012	Approved 1.1 Version
1.2.0	21 July 2013	Approved 1.2 Version
1.1.1	21 July 2013	Approved 1.1 Errata Version
1.1.2	04 September 2013	Approved 1.1 Errata Version
1.1.3	05 November 2013	Approved 1.1 Errata Version
1.1.4	18 March 2014	Approved 1.1 & 1.2 Errata Version
1.1.5	14 May 2014	Approved 1.1 Errata Version
1.1.6	29 May 2014	Approved 1.1 Errata Version
1.1.7	18 March 2015	Approved 1.1 Errata Version
1.2.2	17 June 2015	Approved 1.1 Errata Version

# LIST OF CONTRIBUTORS

2

3

4 Brakensiek, Jörg (Editor) Microsoft Corporation

5 Soundararajan, Murali Samsung

6 Lünnemann, Patrick Carmeq GmbH/Volkswagen AG

## **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, the Developer
- 45 Agreement found on the Developer Portal ("Developer Agreement") and the CCC LLC Limited Liability Company Agreement
- (the "LLC Agreement").

1

11

- 6 Use of the Specification by anyone who is not a registered developer ("Developer") or a member of the CCC LLC (each
- such person or party, a "Member") is prohibited. The legal rights and obligations of Developers are governed by the Devel-
- oper Agreement found on the Developer Portal. The legal rights and obligations of each Member are governed by the Car
- Connectivity Consortium LLC Agreement and their applicable Membership Agreement, including without limitation those
- 10 contained in Article 10 of the LLC Agreement.

#### FOR MEMBERS AND DEVELOPERS

- 12 CCC LLC hereby grants each Member and Developer a right to use and to make verbatim copies of the Specification for
- 13 the purposes of implementing the technologies specified in the Specification in their products ("Implementing Products")
- 14 under the terms of the LLC Agreement or Developer Agreement, as appropriate (the "Purpose"). No other license, express
- 15 or implied, by estoppel or otherwise, to any intellectual property rights are granted herein.
- THE SPECIFICATION IS PROVIDED "AS IS" WITH NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT
- 17
- LIMITATION ANY WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS, AND COMPLIANCE WITH 18
- 19 APPLICABLE LAWS.
- 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 PROPERTY RIGHTS OR FOR NONCOMPLIANCE WITH LAWS, RELATING TO USE OF THE
- 20 21 22 23 24 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.
- 25 CCC LLC reserves the right to adopt any changes or alterations to the Specification as it deems necessary or appropriate.
- Each Member or Developer, as appropriate, (i) hereby acknowledges that its Implementing Products may be subject to
- 26 27 28 29 30 31 32 33 34 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, telecommu-
- nications regulations, technology transfer controls and health and safety regulations, (ii) is solely responsible for the com-
- pliance by their Implementing Products with any such laws and regulations and for obtaining any and all required authoriza-
- tions, permits, or licenses for their Implementing Products related to such regulations within the applicable jurisdictions, and
- (iii) acknowledges that nothing in the Specification provides any information or assistance in connection with securing such
- compliance, authorizations or licenses.

#### 35 FOR DEVELOPERS ONLY

- Any use of the Specification not in compliance with the terms of this Legal Notice and the Developer Agreement is prohibited
- 36 37 38 and any such prohibited use may result in termination of the Developer Agreement and in other liability as permitted by the
- Developer Agreement or by applicable law to the CCC LLC or any of its Members for patent, copyright and/or trademark
- 39 infringement. Developers are not permitted to make available or distribute this Specification or any copies thereof to any
- 40 third party.

#### FOR MEMBERS ONLY

- 42 Any use of the Specification not in compliance with the terms of this Legal Notice, the LLC Agreement, and the Membership
- 43 Agreement is prohibited and any such prohibited use may result in termination of the applicable Membership Agreement
- 44 and in other liability as permitted by the such Membership Agreement or by applicable law to the CCC LLC or any of its
- 45 Members for patent, copyright and/or trademark infringement.
- 46 This Specification may not be provided to any third party other than to Affiliates of Members (as defined in the LLC Agree-
- 47 ment) and subcontractors but only to the extent that such Affiliates and subcontractors have a need to know for carrying out
- 48 the Purpose and provided that such Affiliates and subcontractors accept confidentiality obligations similar to those contained
- 49 in the LLC Agreement. Each Member shall be responsible for the observance and proper performance by such of its Affiliates
- 50 and subcontractors of the terms and conditions of this Legal Notice and the LLC Agreement.
- 51 Copyright © 2011-2015. CCC LLC.

52

# **TABLE OF CONTENTS**

LIST OF CONTRIBUTORS	2
LEGAL NOTICE	3
TABLE OF CONTENTS	4
3 DEFINITIONS	10
3.2 0xE002 – Structure ServiceInfo	10
3.3 0xE003 – Structure Action	10
3.4 0xE004 – FBCONTEXT	10
4 COMMON API ELEMENTS	12
The once of Editional Remote English Confidence in the contraction in the contract of the cont	
4 5 2 0x0402 – Display Configuration Callback	22
4.5.3 0x0403 – Client Pixel Format	23
***	
4.6.1 0x0501 – Event Configuration	25
4.6.4 0x0504 – Get Event Mapping	26
4.7 0x06xx – Client Virtual Keyboard	28
	3.1

1	4.7.1 0x0601 – Show Client Virtual Keyboard	28
2	4.7.2 0x0602 – Client Virtual Keyboard Support	28
3	4.7.3 0x0603 – Client Virtual Keyboard Text Entry Callback	28
4	4.8 0x07xx – Key Event Listing	
5	4.8.1 0x0701 – Key Event List	
6	4.8.2 0x0702 – Key Event List Support	
7	4.9 0x08xx – Context Information	
8	4.9.1 0x0801 – Framebuffer Context Information	
9	4.9.2 0x0802 – Framebuffer Blocking Information Callback	
10	4.9.3 0x0803 – Audio Context Information	
11	4.9.4 0x0804 – Audio Blocking Information Callback	
12	4.9.5 0x0805 – Framebuffer Unblocking Callback	
13	4.9.6 0x0806 – Audio Unblocking Callback	34
14	4.10 0x09xx – Device Status Information	
15	4.10.1 0x0901 – Drive Mode	
16	4.10.2 0x0902 – Drive Mode Callback	
17	4.10.3 0x0903 – Night Mode	
18	4.10.4 0x0904 – Night Mode Callback	
19	4.10.5 0x0905 – Microphone State	
20	4.10.6 0x0906 – Open Microphone Callback	
21	4.10.7 0x0907 – Set Open Microphone	
22	4.11 0x0Axx – Data Services	
23	4.11.1 0x0A01 – Get Available Services	
24	4.11.2 0x0A02 – Available Services Callback	37
25	4.11.3 0x0A03 – Register to a Service	
26	4.11.4 0x0A04 – Register to a Service Callback	
27	4.11.5 0x0A05 – Unregister from a Service	
28	4.11.6 0x0A06 – Subscribe to an Object	
29	4.11.7 0x0A07 – Subscribe to an Object Callback	
30	4.11.8 0x0A08 – Unsubscribe from an Object	
31	4.11.9 0x0A09 – Set an Object	
32	4.11.10 0x0A0A – Set Object Callback	
33	4.11.11  0x0A0B – Get an Object	
34	4.11.12 0x0A0C – Get Object Callback	
35	4.12 0x0Bxx – Notifications	
36	4.12.1 0x0B01 – Notifications Supported	
37	4.12.2 0x0B02 – Notifications Enabled	
38	4.12.3 0x0B03 – Notifications Enabled Callback	
39	4.12.4 0x0B04 – Notification Configuration	
40	4.12.5 0x0B05 – Notification Configuration Callback	
41	4.12.6 0x0B06 – Send Notification for client-based Notification UI	
42	4.12.7 0x0B07 – Send Notification for VNC-based Notification UI	
43	4.12.8 0x0B08 - Cancel Notification	
44 45	4.12.9 0x0B09 – Receive Action Callback	
45 46	4.13 0x0Cxx – Web Application specific Methods	
46 47	4.14 0x0Dxx – Misc. Mirror Link 1.2 Additions	
47 40	4.14.1 0x0D01 – MirrorLink Client Driver Distraction Information	
48	4.14.2 0x0D02 – MirrorLink Client Driver Distraction Callback	43
49	5 REFERENCES	46

# TERMS AND ABBREVIATIONS

2	ACMS	Application Certification Management System
3	BT	Bluetooth
4	ML	MirrorLink
5	OCSP	Online Certificate Status Protocol
6	RFB	Remote Framebuffer
7	UPnP	Universal Plug and Play
8	USB	Universal Serial Bus
9	VNC	Virtual Network Computing
10		

- 11 MirrorLink is a registered trademark of Car Connectivity Consortium LLC
- 12 Bluetooth is a registered trademark of Bluetooth SIG Inc.
- 13 RFB and VNC are registered trademarks of RealVNC Ltd.
- 14 UPnP is a registered trademark of UPnP Forum.
- Other names or abbreviations used in this document may be trademarks of their respective owners.

# 1 ABOUT

1

10

1112

13

14

15

16

17 18

19 20

21

22

23

24

25

26

27

- 2 This document specifies the features of the MirrorLink Common API, available for all MirrorLink Certified
- 3 Applications on a MirrorLink Certified Server device.
- 4 The specification lists a series of requirements, either explicitly or within the text, which are mandatory ele-
- 5 ments for a compliant solutions. Recommendations are given, to ensure optimal usage and to provide suitable
- 6 performance. All recommendations are optional.
- 7 The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD",
- 8 "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are following the no-
- 9 tation 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 Introduction

1

12

13

14

15

17

18 19

- 2 The Common API specifies an interface to the MirrorLink Server, which allows any application to either get
- 3 information about MirrorLink Server's or Client's properties or to set them to specific values. In addition, the
- 4 API specifies callback functions, which are used from the MirrorLink Server to inform the application about
- 5 any change. Callback functions MUST be implemented from the applications for any evented function.
- 6 The Common API specifies the interface in a platform/OS independent manner. Platform specific specifica-
- 7 tion will describe the detailed platform specific view of the Common API, which MUST be implemented
- 8 from any MirrorLink Server device.
- The platform specific implementation of the Common API MUST provide method to implement the features specified in this document, with at least the values provided in this specification.
- 11 A specific API function can be marked as Mandatory or Optional.
  - Any Mandatory marked function MUST be fully implemented from the MirrorLink Server
  - Any Optional marked function SHOULD be fully implemented from the MirrorLink Server. In case
    the function is not fully implemented, the MirrorLink Server MUST implement an empty shell,
    which responds with defined default values and a success flag set to "False" (if available).
- 16 The Common API specifies functions with three types of API functions:
  - Get: The function is providing read access to information available on the MirrorLink Server.
  - Set: The function is providing write access to information available on the MirrorLink Server.
  - Callback: The function is a callback function, invoked from the MirrorLink Server. The implementation of the callback functionality will be specified in the platform specific specifications.
- All three functions may have a Success return value specified. The return value is set to True, if the action has been successful or the information requested is available. Otherwise the return value is set to False.
- Some of the data provided via the Common API will not be available from MirrorLink 1.0 clients. In such case, the MirrorLink Server MUST provide a default value as specified.
- 25 The Common API uses a set of Data Types, given in the table below. The platform specific API MAY use
- other data types, as long as the original intend of the Common API is not compromised. Therefore the plat-
- 27 form specific implementation of the Common API MAY use existing platform APIs are sub-classed versions
- of them.

Data Type	Description
	Data type representing the logical values true and false.
bool	The representation of false is all-bits-zero, and the representation of true is unspecified except that it shall have at least one bit set.  Default: FALSE
uint8	Data type representing integer values ranging from 0 to positive 255 (0xFF)  Default: 0
uint16	Data type representing integer values ranging from 0 to positive 65,535 (0xFFFF)  Default: 0
uint32	Data type representing integer values ranging from 0 to positive 4,294,967,295 (0xFFFFFFFF)  Default: 0
int8	Data type representing integer values ranging from negative 128 (0x80) to positive 127 (0x7F)  Default: 0

Data Type	Description	
int16	Data type representing integer values ranging from negative 32,768 (0x8000) to positive 32,767 (0x7FFF)  Default: 0	
int32	Data type representing an integer values ranging from negative 2,147,483,648 (0x80000000) to positive 2,147,483,647 (0x7FFFFFFF)  Default: 0	
float	Data type representing a 32 bit floating point value according IEEE754-1985, single-precision  Default: 0.0	
double	Data type representing a 64 bit floating point value according IEEE754-1985, double-precision Default: 0.0	
string8	Array of UTF8 characters. Each character takes 1 byte (UTF8).  Default: ""	
string16	Array of UTF16 characters. Each character takes 2 bytes (UTF16).  Default: ""	
url	Data type representing a URL Default: ""	
typeName[]	Data type representing an array of values of type typeName.  Defalut: Zero-length array	
structure- Name	Data type representing the Structure <i>typeName</i> , as specified in Chapter Definitions.  Default: Default value for each element of the structure	
void*	Pointer to a data structure  Default: "0x0"	

Table 1: Data Types and Default Values

- 2 The Common API does not intend to specify, how information provided the MirrorLink has to be used to
- 3 fulfill driver distraction guidelines. This information is provided from driver distraction guideline documents
- 4 and associated test plans.

- 5 If the Common API replicates functionality, available via OS/Platform APIs, then those API MUST be used,
- 6 as defined in the Platform specific specifications.
- 7 The platform specific API MAY rearrange the defined parameter, or add additional parameter. The platform
- 8 specific API MUST NOT remove any parameter.

# 3 DEFINITIONS

# 2 3.1 0xE001 - Structure Rect

Feature Name	Description	Type
x	Horizontal offset of the upper left corner	uint16
У	Vertical offset of the upper left corner	uint16
width	Width of the rectangle	uint16
height	Height of the rectangle	uint16

Table 2: Structure Rect

# 4 3.2 0xE002 - Structure ServiceInfo

Feature Name	Description	Type
Minor Version	Minor service version	uint8
Major Version	Major service version	uint8
Service ID	Service identifier	uint16
Name	Service name	string8

Table 3: Structure ServiceInfo

# 6 3.3 0xE003 - Structure Action

Feature Name	Description	Type
actionID	Action identifier; MUST be non-zero. The actionIDs MUST be unique within one notification. Otherwise the MirrorLink Server will reject the notification.	
name	Action name	string8
launchApp  Flag whether to launch the app  Default: False		bool
iconUrl  URL to the icon associated with the action Icon MUST be of mimetype "image/png" with a color depth of 24.  Default: No Icon		url

Table 4: Structure Action

# 8 3.4 0xE004 - FbContext

Feature Name	Description	Type
applicationCategory	Category of the application	uint32
videoContentCate- gory	Category of the framebuffer video content.	uint32
framebufferArea	Framebuffer rectangle for the specified region.	Rect

7

3

Table 5: Structure FbContext

2



1

# 4 COMMON API ELEMENTS

The MirrorLink common API consists of multiple optional and mandatory modules. Their availability and obligation of a module is dependent on the API level as defined in, as listed in the table below.

Common API Module	Module Reference	API Level 0x01	API Level 0x02
Common API Info	0xF0	Mandatory	Mandatory
Device Info	0x01	Mandatory	Mandatory
Certification Information	0x02	Mandatory	Mandatory
Connection Information	0x03	Mandatory	Mandatory
Display Information	0x04	Mandatory	Mandatory
Event Information	0x05	Mandatory	Mandatory
Client Virtual Keyboard	0x06	Optional	Optional
Key Event Listing	0x07	Optional	Optional
Context Information	0x08	Mandatory	Mandatory
Device Status Information	0x09	Mandatory	Mandatory
Data Services	0x0A	Optional	Optional
Notifications	0x0B	Optional	Optional
Web Applications	0x0C	Not available	Reserved for future use
Misc. ML 1.2 Features	0x0D	Not available	Mandatory

Table 6: Common API Modules

- 5 Any MirrorLink Server MUST implement all mandatory modules and all functions within that module. Any
- application using the Common API MUST implement all given Callback functions required for the operation of the application; the platform specific specification MAY provide conditions for the obligation of individual
- 8 callback functions.

- 9 Any MirrorLink Server MUST implement all functions within an optional module, if it supports that module.
- 10 The MirrorLink Server MUST provide a mechanism to check, whether a module is available. Any application
- using an optional module of the Common API MUST implement all given Callback functions required for
- 12 the operation of the application; the platform specific specification MAY provide conditions for the obligation
- of individual callback functions.
- 14 The MirrorLink applications MUST use the 0x0301 Common API Call and the 0x0302 Common API
- 15 Callback to determine, whether a MirrorLink session in established. MirrorLink applications SHOULD use
- the other Common API modules only, while a MirrorLink session is running.
- 17 MirrorLink Servers MUST have the Common API modules available at all times.

# 1 4.1 0xF0xx - MirrorLink Common API Info

2 4.1.1 0xF001 – MirrorLink Common API Version

3 Description: Implemented MirrorLink Common API Version from the MirrorLink Server

4 Obligation: Mandatory

5 Type: Get

6 Feature List:

Feature Name	Description	Type	Direction
API Level	MirrorLink Common API level	uint16	Read

# 7 4.1.2 0xF002 – Common API Module Available

8 Description: Check, whether MirrorLink Server supports a specific Common API module

9 Obligation: Mandatory

10 Type: Get

Feature Name	Description	Type	Direction
Module Reference	Module reference as defined in Table 6.	uint16	Write
Available	Flag, to indicate whether the module is available	bool	Read

#### 4.2 0x01xx - MirrorLink Device Info

#### 4.2.1 0x0101 – MirrorLink Version

3 Description: Available MirrorLink Version for the established connection, as agreed between the Mir-

rorLink Server and Client. Information MUST be available as soon as the MirrorLink session is connected (refer to 4.4.2); any later change to the provided information MUST

be notified via the callback function defined in 4.2.2.

7 Obligation: Mandatory

8 Type: Get

9 Feature List:

1

2

4 5

6

Feature Name	Description	Туре	Direction
Major Version	MirrorLink major version; return 1 if version information is not available	uint16	Read
Minor Version	MirrorLink minor version; return 0 if version information is not available	uint16	Read
Success	Flag, to indicate whether the information is available	bool	Read

#### 10 4.2.2 0x0102 – MirrorLink Version Callback

11 Description: Indicates that the MirrorLink Version information has changed or became available.

12 Obligation: Mandatory13 Type: Callback

14 Feature List:

<b>Feature Name</b>	Description	Type	Direction
Major Version	MirrorLink major version; return 1 if version information is not available	uint16	Read
Minor Version	MirrorLink minor version; return 0 if version information is not available	uint16	Read

#### 15 4.2.3 0x0103 – MirrorLink Client Manufacturer and Model Information

16 Description: Provided MirrorLink client manufacturer and model information, as received through the

17 UPnP Client Profile Service; any later change to the provided information MUST be no-

tified via the callback function defined in 4.2.4.

19 Obligation: Mandatory

20 Type: Get

21 Feature List:

Feature Name	Description	Type	Direction
Client Identifier	Identifier of the MirrorLink client	string8	Read
Friendly Name	Short user-friendly description of the MirrorLink client	string8	Read
Manufacturer	Manufacturer Name of the MirrorLink client	string8	Read
Model Name	Model name of the MirrorLink client	string8	Read
Model Number	Model number of the MirrorLink client	string8	Read

Feature Name	Description	Type	Direction
Success	Flag, to indicate whether the information is available	bool	Read

#### 1 4.2.4 0x0104 – MirrorLink Client Manufacturer and Model Information Callback

2 Description: Indicates that the Client information has changed.

3 Obligation: Mandatory4 Type: Callback

5 Feature List:

Feature Name	Description	Type	Direction
Client Identifier	Identifier of the MirrorLink client	string8	Read
Friendly Name	Short user-friendly description of the MirrorLink client	string8	Read
Manufacturer	Manufacturer Name of the MirrorLink client	string8	Read
Model Name	Model name of the MirrorLink client	string8	Read
Model Number	Model number of the MirrorLink client	string8	Read

# 4.2.5 0x0105 – Server Device Virtual Keyboard Support

Description: Provides information about the available virtual keyboard from the MirrorLink Server,

which can be used from application, during a MirrorLink session. Handling of the virtual keyboard is following regular platform specific means. Note: The availability of a virtual

keyboard from the MirrorLink Client is covered in section 4.7.

11 Obligation: Mandatory

12 Type: Get

13 Feature List:

6 7

8

9

10

Feature Name	Description	Type	Direction
Available	Flag, to indicate the availability of a virtual keyboard from the MirrorLink Server.	bool	Read
Touch Support	Flag, to indicate whether the virtual keyboard supports touch events.	bool	Read
Knob Support	Flag, to indicate whether the virtual keyboard supports knob events.	bool	Read
Drive Mode	Flag, to indicate whether the virtual keyboard is following driver distraction ruling, as set force for CCC drivecertification	bool	Read

# 1 4.3 0x02xx - Certification Information

#### 2 4.3.1 0x0201 – Get Application Certification Status

3 Description: Provided application certificate status, as captured from the application certificate.

4 Obligation: Mandatory

5 Type: Get

6 Feature List:

Feature Name	Description	Type	Direction
Certificate Available	Flag, indicating whether the MirrorLink server has a valid certificate for the application	bool	Read
Advertised as Certified App	Flag, indicating, whether the MirrorLink server has included the application into its UPnP advertisements as a certified application.	bool	Read

#### 7 4.3.2 0x0202 – Get Application Certifying Entities

8 Description: Provide information on the certifying entities

9 Obligation: Mandatory

10 Type: Get

11 Feature List:

Feature Name	Description	Type	Direction
Entity	Comma-separated list of certifying entities, which certified the application	string8	Read

#### 12 4.3.3 0x0203 – Get Application Certification Information

13 Description: Provided application certificate information; any later change to the provided information

MUST be notified via the callback function defined in 4.3.4,

15 Obligation: Mandatory

16 Type: Get

17 Feature List:

14

18

Feature Name	Description	Туре	Direction
Entity	Name of the certifying entity	string8	Write
Certified	Flag, whether the application has been certified from the given entity	bool	Read
Restricted	Comma-separated list of locales for which the application has been certified for restricted use (drive-level) from the given entity	string8	Read
Non Restricted	Comma-separated list of locales for which the application has been certified for non-restricted use (baselevel) from the given entity	string8	Read

# 4.3.4 0x0204 – Get Application Certification Information Callback

19 Description: Indicate that the application certificate information changed.

Obligation: Mandatory
 Type: Callback

## 3 Feature List:

Feature Name	Description	Type	Direction
Entity	Name of the certifying entity	string8	Write
Certified	Flag, whether the application has been certified from the given entity	bool	Read
Restricted	Comma-separated list of locales for which the application has been certified for restricted use (drive-level) from the given entity	string8	Read
Non Restricted	Comma-separated list of locales for which the application has been certified for non-restricted use (baselevel) from the given entity	string8	Read

#### 4.4 0x03xx - Connection Information

#### 4.4.1 0x0301 - Established MirrorLink Connection

3 Description: Established MirrorLink connection; any later change to the provided information MUST

be notified via the callback function defined in 4.4.2.

5 Obligation: Mandatory

Get 6 Type:

7 Feature List:

1

2

4

8 9

10

11

14 15

16

17

18

19

20

21

22

Feature Name	Description	Type	Direction
Connection	Flag, whether MirrorLink connection has been established.	bool	Read

#### 4.4.2 0x0302 - Established MirrorLink Connection Callback

Indicate that the MirrorLink connection status changed. The callback MUST be provided Description: to all applications, which have registered to the MirrorLink Common API, independent

on whether the application has been launched within or outside a MirrorLink session.

12 A MirrorLink connection is established latest in the following situation (whatever comes 13

first):

MirrorLink Client sends a UPnP SetClientProfile action with an nonempty Client Profile string,

MirrorLink Client sends the first UPnP Application Server service action.

A MirrorLink connection is terminated latest in the following situation (whatever comes

first):

MirrorLink Clients sends a UPnP SetClientProfile action with an empty

Client Profile string,

MirrorLink Server sends a SSDP: byebye message,

Loss of the physical connection (e.g. pulling the USB cable, switching of Wi-Fi)

23 Obligation: Mandatory

24 Type: Callback

25 Feature List:

Feature Name	Description	Type	Direction
Connection	Flag, whether MirrorLink connection has been established.	Bool	Read

#### 4.4.3 0x0303 - Established Audio Connections 26

27 Established Audio connections within MirrorLink setup; any later change to the provided Description:

28 information MUST be notified via the callback function defined in 4.4.4.

29 Obligation: Mandatory

30 Type: Get

Feature Name	Description	Type	Direction
Media Audio Out	Identifier of the audio connection for media audio (output)	uint8	Read

Feature Name	Description	Туре	Direction
	0x01: Not established 0x02: BT A2DP		
	0x03: RTP		
	Identifier of the audio connection for media audio (input)	uint8	Read
Media Audio In	0x00: Not available 0x01: Not established 0x03: RTP		
	Identifier of the audio connection for Voice Control audio (input)	uint8	Read
Voice Control	0x00: Not available 0x01: Not established 0x02: BT HFP + BVRA (Voice Control is outside MirrorLink Server's responsibility; application must use existing platform APIs) 0x03: RTP		
Phone Audio	Identifier of the audio connection for Phone audio (input & output)  0x00: Not available  0x01: Not established  0x02: BT HFP  0x03: RTP	uint8	Read
RTP Payload Types	Comma separated list of supported RTP payload types in case an RTP connection is used.	string8	Read
IPL	Initial Playback Latency value (in ms)  Defines the expected initial latency (e.g. due to audio buffer filling at the MirrorLink client), before any audio is heard via the MirrorLink Client's speaker system.	uint32	Read

# 1 4.4.4 0x0304 – Established Audio Connections Callback

2 Description: Indicate that the audio connections changed.

3 Obligation: Mandatory

4 Type: Callback

Feature Name	Description	Type	Direction
Media Audio Out	Identifier of the audio connection for media audio (output) – see definitions above	uint8	Read
Media Audio In	Identifier of the audio connection for media audio (input) – see definitions above	uint8	Read
Voice Control	Identifier of the audio connection for Voice Control audio (input) – see definitions above	uint8	Read
Phone Audio	Identifier of the audio connection for Phone audio (input & output) – see definitions above	uint8	Read

Feature Name	Description	Type	Direction
RTP Payload Types	Comma separated list of supported RTP payload types in case an RTP connection is used.	string8	Read
	Initial Playback Latency value (in ms)	uint32	Read
IPL	Defines the expected initial latency (e.g. due to audio buffer filling at the MirrorLink client), before any audio is heard via the MirrorLink Client's speaker system.		

## 4.4.5 0x0305 – Established Remote Display Connection

2 Description: Established remote display connection; any later change to the provided information

MUST be notified via the callback function defined in 4.4.6.

4 Obligation: Mandatory

5 Type: Get

6 Feature List:

1

Feature Name	Description	Туре	Direction
	Identifier of the remote display type. Must uniquely identify the following types	uint8	Read
Connection	0x00: No connection established 0x01: VNC 0x02: HSML (MirrorLink ≥ 1.2 only) 0x03: WFD (MirrorLink ≥ 1.2 only) 0xFF: Other		

# 7 4.4.6 0x0306 - Established Remote Display Connection Callback

8 Description: Indicate that the remote display connections changed.

9 Obligation: Mandatory

10 Type: Callback

11 Feature List:

Feature Name	Description	Type	Direction
Connection	Identifier of the remote display type. Must uniquely identify the following types – see definitions above	uint8	Read

# 4.5 0x04xx – Display Information

The Common API does not specify how information provided the MirrorLink has to be used to fulfill driver distraction.

#### 4.5.1 0x0401 – Display Configuration

Description: Access information on the display properties of the MirrorLink Session; this information is used by MirrorLink certified applications to adapt its user interface to fulfill driver distraction guidelines, in particular regarding font sizes; Requires an established VNC connection; any later change to the provided information MUST be notified via the callback function defined in 4.5.2.

The provided framebuffer resolutions are modeling the following framebuffer pipeline:

- 1. The applications renders its user interface into a framebuffer available in full to the application (App Horizontal / Vertical Resolution)<sup>1</sup>
- 2. The MirrorLink Server scales that framebuffer to better fit the MirrorLink Client's framebuffer properties (Server Horizontal / Vertical Resolution)
- 3. The MirrorLink Server adds pad rows and/or columns to the scaled framebuffer (Server Pad Rows / Columns)
- 4. The MirrorLink Server transmits that framebuffer to the MirrorLink Client
- 5. The MirrorLink Client scales the received framebuffer to fit into its framebuffer (Client Horizontal / Vertical Resolution); the MirrorLink Client may add pad rows or columns (but not both) to compensate for differences in the framebuffer aspect ratio. Those pad rows or columns to not take away any resolution from the transmitted MirrorLink Server framebuffer.

All pixel-based resolutions MUST be based on a pixel aspect ratio of 1 (one), i.e. a squared pixel.

25 Obligation: Mandatory

26 Type: Get

#### 27 Feature List:

1

4 5

6 7

8

10

11 12

13

14

15

16

17

18

19 20

21

2223

24

\_

Feature Name	Description	Type	Direction
App Horizontal Resolution	Horizontal resolution in pixel of the framebuffer, the application is rendering into.	uint16	Read
	Note: In many cases, the App Horizontal Resolution equals the horizontal resolution of the MirrorLink Server's display.		
App Vertical Resolution	Vertical resolution in pixel of the framebuffer, the application is rendering into.  Note: In many cases, the App Vertical Resolution equals the vertical resolution of the MirrorLink Server's display.	uint16	Read
Server Horizon- tal Resolution	Horizontal resolution in pixel, after the MirrorLink Server has scaled the application framebuffer.	uint16	Read

<sup>&</sup>lt;sup>1</sup> If the application is using the MirrorLink Server's physical framebuffer, then the App Horizontal / Vertical Resolution is the resolution of the MirrorLink Server Device Display.

Feature Name	Description	Type	Direction
Server Vertical Resolution	Vertical resolution in pixel, after the MirrorLink Server has scaled the application framebuffer.	uint16	Read
Server Pad Rows	Number of pad rows added from the MirrorLink Server to the scaled application framebuffer	uint16	Read
Server Pad Col- umns	Number of pad columns added from the MirrorLink Server to the scaled application framebuffer	uint16	Read
Client Horizon- tal Resolution	Horizontal resolution in pixel of the MirrorLink Client framebuffer, available for rendering the MirrorLink Server's screen.	uint16	Read
Client Vertical Resolution	Vertical resolution in pixel of the MirrorLink Client framebuffer, available for rendering the MirrorLink Server's screen	uint16	Read
Width	Physical width in mm of the MirrorLink Client display, where the MirrorLink Server's screen appears.	uint16	Read
Height	Physical height in mm of the MirrorLink Client display, where the MirrorLink Server's screen appears.	uint16	Read
Distance	Physical distance in mm of the MirrorLink Client display from the driver's head position.	uint16	Read
App Pixels Per Client mm			Read
Success	Flag, to indicate whether the information is available	bool	Read

# 1 4.5.2 0x0402 – Display Configuration Callback

2 Description: Display Configuration has changed.

3 Obligation: Mandatory4 Type: Callback

Feature Name	Description	Type	Direction
App Horizontal Resolution	Horizontal resolution in pixel of the framebuffer, the application is rendering into.	uint16	Read
	Note: In many cases, the App Horizontal Resolution equals the horizontal resolution of the MirrorLink Server's display.		
App Vertical Resolution	Vertical resolution in pixel of the framebuffer, the application is rendering into.	uint16	Read
	Note: In many cases, the App Vertical Resolution equals the vertical resolution of the MirrorLink Server's display.		
Server Horizon- tal Resolution Horizontal resolution in pixel, after the MirrorLink Server has scaled the application framebuffer.		uint16	Read

Feature Name	Description	Type	Direction
Server Vertical Resolution	Vertical resolution in pixel, after the MirrorLink Server has scaled the application framebuffer.	uint16	Read
Server Pad Rows	Number of pad rows added from the MirrorLink Server to the scaled application framebuffer	uint16	Read
Server Pad Col- umns	Number of pad columns added from the MirrorLink Server to the scaled application framebuffer	uint16	Read
Client Horizon- tal Resolution	Horizontal resolution in pixel of the MirrorLink Client framebuffer, available for rendering the MirrorLink Server's screen.	uint16	Read
Client Vertical Resolution	Vertical resolution in pixel of the MirrorLink Client framebuffer, available for rendering the MirrorLink Server's screen	uint16	Read
Width	Physical width in mm of the MirrorLink Client display, where the MirrorLink Server's screen appears.	uint16	Read
Height	Physical height in mm of the MirrorLink Client display, where the MirrorLink Server's screen appears.	uint16	Read
Distance	Physical distance in mm of the MirrorLink Client display from the driver's head position.	uint16	Read
App Pixels Per Client mm	Number of application-level pixels, which will fit into 1 mm of Client Display space.  Note: This value is the same for the horizontal and vertical dimension.	float	Read

#### 4.5.3 0x0403 – Client Pixel Format

2 Description: Access information about the pixel format of the framebuffer data, being transmitted to

the MirrorLink Client; requires established VNC connection; any later change to the pro-

vided information MUST be notified via the callback function defined in 4.5.4.

5 Obligation: Mandatory

6 Type: Get

7 Feature List:

1

3

4

8

Feature Name	Descri	ption	Type	Direction		
	Pixel fo	rmat value, as	given belo	w:	uint8	Read
	0x01:	ARGB888	0x05:	RGB444		
Pixel Format	0x02:	RGB888	0x06:	RGB343		
	0x03:	ARGB565	0x07:	16-Bit-Gray		
	0x04:	RGB555	0x08:	8-Bit-Gray		
Success	Flag, to	Flag, to indicate whether the information is available				Read

#### 4.5.4 0x0404 – Client Pixel Format Callback

9 Description: Pixel format has changed.

10 Obligation: Mandatory11 Type: Callback

#### 1 Feature List:

Feature Name	Description	Type	Direction
Pixel Format	Pixel format value, as given below – see definition above.	uint8	Read

#### 2 4.5.5 0x0405 – Set Framebuffer Orientation Support

3 Description: Inform the MirrorLink Server about the application's framebuffer orientation support; un-

less otherwise set by the application, the VNC Server MUST assume that the application

will only support Landscape.

6 Obligation: Mandatory

7 Type: Set

8 Feature List:

Feature Name	Description	Type	Direction
	Orientation of the Application Framebuffer Unique values for at least the following configurations:	uint8	Write
Framebuffer Orientation	0x01: Landscape only ( <b>default</b> ) 0x02: Portrait only 0x03: Landscape and Portrait		
Success	Flag, indicating whether the action is successful.	bool	Read

9

14

15

16

4

5

- 10 The application MUST use platform specific APIs to switch its framebuffer orientation. If the new orientation
- is not supported from the MirrorLink client, the application will receive a Switch Framebuffer Orientation
- 12 Callback as specified in 4.5.6.

#### 13 4.5.6 0x0406 – Switch Framebuffer Orientation Callback

Description: MirrorLink Server requests a framebuffer orientation switch from the application. The

actual switch will happen via regular OS/platform mechanisms. An application MUST

switch its orientation, if it has indicated support for Landscape and Portrait in chapter 0.

17 Obligation: Mandatory

18 Type: Callback

19 Feature List:

Feature Name	Description	Type	Direction
Framebuffer Orientation	Requested orientation of the Application Framebuffer true: Landscape false: Portrait	bool	Read

#### 4.6 0x05xx – Event Information

# 4.6.1 0x0501 – Event Configuration

Description: Access information on the event properties of the MirrorLink connection, i.e. the event

properties, which are supported from both, the MirrorLink Server and MirrorLink Client; details on the event configuration are specified in the VNC specification; Requires established the ANNO of TH

lished VNC connection; any later change to the provided information MUST be notified

via the callback function defined in 4.6.2.

8 Obligation: Mandatory

9 Type: Get

10 Feature List:

1

2

3

4

5

6

7

Feature Name	Description	Type	Direction
Knob Support	Supported knob events from the MirrorLink Server and Client. Bit mask as defined in the VNC specification.	uint32	Read
Device Key Sup- port	Supported device key events from the MirrorLink Server and Client. Bit mask as defined in the VNC specification.	uint32	Read
Multimedia Key Support	Supported multimedia key events from the MirrorLink Server and Client. Bit mask as defined in the VNC specification.	uint32	Read
Function Key Support	Number of supported function keys from the MirrorLink Server and Client.	uint8	Read
ITU Key Support	Support for ITU keys from the MirrorLink Server and Client	bool	Read
Touch event support	Number of simultaneous touch events, supported from the MirrorLink Server and Client:  0x00: No touch support  0x01: Single-Touch events only  Other: Multi-Touch support (Gestures)	uint8	Read
Pressure Mask	The pressure mask indicates how many pressure levels can be distinguished from the MirrorLink Server and Client.	uint8	Read
Keyboard Language	Language & country codes for Virtual Keyboard setting at the MirrorLink Client, e.g. "en/us"	string8	Read
UI Language	Language & country codes for UI Language setting at the MirrorLink Client, e.g. "en/us"	string8	Read
Success	Flag, to indicate whether the information is available	bool	Read

## 11 4.6.2 0x0502 – Event Configuration Callback

12 Description: Client event configuration information has changed.

13 Obligation: Mandatory14 Type: Callback

Feature Name	Description	Type	Direction
Knob Support	Supported knob events from the MirrorLink Server and Client. Bit mask as defined in the VNC specification.	uint32	Read
Device Key Sup- port	Supported device key events from the MirrorLink Server and Client. Bit mask as defined in the VNC specification.	uint32	Read
Multimedia Key Support	Supported multimedia key events from the MirrorLink Server and Client. Bit mask as defined in the VNC specification.	uint32	Read
Function Key Support	Number of supported function keys from the MirrorLink Server and Client.	uint8	Read
ITU Key Support	Support for ITU keys from the MirrorLink Server and Client	bool	Read
Touch event support	Number of simultaneous touch events, supported from the MirrorLink Server and Client – see definitions above	uint8	Read
Keyboard Lan- guage	Language & country codes for Virtual Keyboard setting at the MirrorLink Client, e.g. "en/us"	string8	Read
UI Language	Language & country codes for UI Language setting at the MirrorLink Client, e.g. "en/us"	string8	Read
Pressure Mask	The pressure mask indicates how many pressure levels can be distinguished from the MirrorLink Server and Client.	uint8	Read

# 4.6.3 0x0503 – Get Remapped Events

Description: Mapping MirrorLink Client events to local MirrorLink Server events; this API call gives

access to the MirrorLink Client events, which are internally mapped to a different local MirrorLink Server event than specified in the Platform Specific Specification; requires an established VNC connection; an application MUST use the function described in 4.6.4 to retrieve the mapping information; any later change to the provided information MUST be

notified via the callback function defined in 4.6.5

Obligation: Mandatory

9 Type: Get

10 Feature List:

1 2

3

4 5

6

7

8

11

13 14

15

Feature Name	Description	Type	Direction
Event List	Array of MirrorLink Client key events, which are mapped within the MirrorLink Server to a different key symbol value than specified within the Platform Specific Specification.	uint32[]	Read
Success	Flag, to indicate whether the information is available	bool	Read

#### 4.6.4 0x0504 – Get Event Mapping

12 Description: Mapping MirrorLink Client events to local MirrorLink Server events; this API call gives

access to the internal mapping in the MirrorLink Server; Requires established VNC connection; any later change to the provided information MUST be notified via the callback

function defined in 4.6.5

16 Obligation: Mandatory

1 Type: Get

#### 2 Feature List:

Feature Name	Description	Type	Direction
Remote Event	Key event value of the remote event	uint32	Write
Local Event	Key event value of the local event, as it will be emulated on the MirrorLink Server device in response to the received remote event.  Will be Zero if no mapping is implemented	uint32	Read
Success	Flag, to indicate whether the information is available	bool	Read

# 3 4.6.5 0x0505 – Get Event Mapping Callback

4 Description: The application MUST be notified, whenever the MirrorLink Server or Client changes a

mapping.

6 Obligation: Mandatory7 Type: Callback

8 Feature List:

Feature Name	Description	Туре	Direction
Remote Event	Key event value of the remote event, which got changed	uint32	Read
Local Event	Key event value of the local event, as it will be emulated on the MirrorLink Server device in response to the received remote event.  Will be Zero if no mapping is implemented	uint32	Read

# 4.7 0x06xx - Client Virtual Keyboard

2 4.7.1 0x0601 – Show Client Virtual Keyboard

3 Description: Trigger a virtual keyboard at the MirrorLink Client; requires an established VNC connec-

tio

5 Obligation: Conditional – Virtual Keyboard Module available

6 Type: Set

7 Feature List:

1

4

Feature Name	Description	Туре	Direction
Virtual Key- board Flag	Flag, to identify whether to show or a remove a virtual keyboard	bool	Write
Text Entry	Text entry, to be used from the virtual keyboard	string16	Write
Key Event List	A key event list is provided separately (after this call)	bool	Write

# 8 4.7.2 0x0602 - Client Virtual Keyboard Support

9 Description: Check, whether MirrorLink client and server support virtual keyboard

10 Obligation: Conditional – Virtual Keyboard Module available

11 Type: Get

12 Feature List:

Feature Name	Description	Type	Direction
Virtual Key- board Flag	Flag to indicate whether a virtual keyboard is supported; must be set to FALSE, if no VNC connection is established.	bool	Read
Text Entry Flag	Flag to indicate whether text entry exchange is supported	bool	Read
Text length	Maximum length of text entry. A value of 0 indicates no constraint.	uint8	Read

#### 13 4.7.3 0x0603 - Client Virtual Keyboard Text Entry Callback

14 Description: Provide completed text entry; this callback is used when the text entry is completed on the

15 MirrorLink Client

16 Obligation: Conditional – Application uses Virtual Keyboard Module

17 Type: Callback

Feature Name	Description	Type	Direction
Text Entry	Text entry, as completed from the virtual keyboard on the MirrorLink Client.	string16	Read

# 1 4.8 0x07xx - Key Event Listing

2 4.8.1 0x0701 - Key Event List

3 Description: Provide a white list of key events; key events are following the MirrorLink client device

4 language setting; requires established VNC connection

5 Obligation: Conditional – Key Event Listing Module available

6 Type: Set

7 Feature List:

Feature Name	Description	Type	Direction
Key Event List	List of supported key events (full white list)	uint32[]	Write

# 8 4.8.2 0x0702 - Key Event List Support

9 Description: Check, whether MirrorLink client and server support key event listing; requires estab-

10 lished VNC connection

11 Obligation: Conditional – Key Event Listing Module available

12 Type: Get

Feature Name	Description	Type	Direction
Key Event List- ing Flag	Flag to indicate whether key event listing is supported; must be set to FALSE, if no VNC connection is established.	bool	Read

#### 4.9 0x08xx - Context Information

#### 4.9.1 0x0801 – Framebuffer Context Information

Provides information of the current framebuffer context; the MirrorLink Description: Server MUST use the application and content category values from the UPnP advertisements, unless otherwise stated from the application using this SET function. The MirrorLink Server MUST use the latest values until a new SET function call is issued. Unless set by the application, the MirrorLink Sever MUST treat the "Handle Blocking"

flag as being set to a FALSE value.

9 Obligation: Mandatory

10 Set Type:

#### 11 Feature List:

1

2

3

4

5

6

7 8

12

14 15

16

20

24

25

26 27

28

29

30

31

Feature Name	Description	Type	Direction
Framebuffer Context Infor- mation	Framebuffer context information  Setting the value to a Zero pointer will reset the video content category to the value provided in the UPnP application advertisement.	fbContext[]	Write
Handle Blocking	Flag, whether the application will take care of the blocking, in case the MirrorLink Client blocks the content.	bool	Write

#### 4.9.2 0x0802 - Framebuffer Blocking Information Callback

13 Description: Framebuffer is blocked from the MirrorLink Client; in case the application has indicated

that it will handle the blocking (refer to 4.9.1) it MUST remove the blocked content.

The MirrorLink Server will handle the Framebuffer Blocking, if the application is unable

to handle the callback in time. This MAY include terminating the application.

17 Obligation: Mandatory

18 Type: Callback

#### 19 Feature List:

Feature Name	Description	Type	Direction
Framebuffer Area	Framebuffer rectangle for the specified region.	Rect	Read
Blocking Rea- son	Reason for blocking	uint16	Read

MirrorLink specified a set of framebuffer blocking reasons, which are provided from the MirrorLink Client. 21 22

Some of these framebuffer blocking notifications will be handled directly from the MirrorLink Server itself,

23 without notifying the application, whereas others are provided to the applications for their further handling.

The following overview lists the blocking reasons specified by MirrorLink and how they are handled from the MirrorLink Server. The MirrorLink Server SHOULD only pass the notification to the application, if no reason flag is set to handle the blocking by itself.

#### Bit 0 – Not allowed content category

The MirrorLink Server MUST send a callback, if the application has previously set the framebuffer context information via the Common API function 0x0801 with the "Handle Blocking" parameter set to TRUE. Otherwise the MirrorLink Server does not know, whether the application can handle the blocking and hence no notification will be sent.

The MirrorLink Server MUST continue sending the callback for a limited time, in case of a CCC certified application. Otherwise the MirrorLink Server MUST handle the situation and no further blocking message is send.

#### • Bit 1 – Not allowed application category

Same behavior as with Bit 0

4

5

6

7 8

9

10

11 12

13

14

15

16

17

18

19

20

21

2526

27

28

29

30

31

32

33

34

#### • Bit 2 – Not sufficient content trust level

MirrorLink Server MUST handle the blocking and no framebuffer blocking notification is sent.<sup>2</sup>

#### • Bit 3 – Not sufficient application trust level

Same behavior as with Bit 2

#### • Bit 4 – Content rules not followed

Same behavior as with Bit 2

#### • Bit 5 – Not allowed application ID

MirrorLink Server MUST handle the blocking and no framebuffer blocking notification is sent.<sup>3</sup>

#### • Bit 8 – UI not in focus on remote display

The MirrorLink Server MUST pass the notification to the application. This notifies the application that the user currently cannot interact with the application using touch and/or knob events, but the application is still visible.

#### • Bit 9 – UI not visible on remote display

The MirrorLink Server MUST pass the notification to the application. This notifies the application that the user cannot see the application on the MirrorLink Client's display.

#### • Bit 10 – UI layout not supported (after a Desktop Size Pseudo Encoding)

- 22 MirrorLink Server MUST handle the blocking and no framebuffer blocking notification is sent.<sup>4</sup>
- A MirrorLink Server, handling a framebuffer blocking notification MUST either put the application into the
- background, terminate it or request the MirrorLink Client to switch to its native user interface.

#### 4.9.3 0x0803 – Audio Context Information

Description: Provides information of the current audio context and whether the application is currently providing audio; The MirrorLink Server MUST use the application category value from the UPnP advertisements, unless otherwise stated from the application using this SET function. The MirrorLink Server MUST use the given values until a new SET function call is issued. Unless set by the application, the MirrorLink Sever MUST treat the "Handle Blocking" flag as being set to a FALSE value.

The application MUST continue updating the information, whenever the context changes, even when the audio is blocked (0x0804) by the MirrorLink Client. The MirrorLink Server MUST store the latest update and use it, whenever needed.

35 Obligation: Mandatory

36 Type: Set

Feature Name	Description	Type	Direction
Audio Content	Application is providing Audio content	bool	Write

<sup>&</sup>lt;sup>2</sup> The application is a non-certified application.

<sup>&</sup>lt;sup>3</sup> The MirrorLink Client uses this reason flag, if it blocks an application for certification status reason.

<sup>&</sup>lt;sup>4</sup> The MirrorLink Server SHOULD use the Switch Framebuffer Orientation Callback (0x0406).

Feature Name	Description	Туре	Direction
	If set to True, the application is going to start an audio stream.		
	If set to False, the application has stopped the audio stream.		
	Array of Application Categories for the Audio Content of the audio stream.	uint32[]	Write
Audio Content Category	Array MUST be sorted in priority order. Top priority is at position [0].		
cutegory	Setting the value to a Zero pointer will reset the audio content category to the value provided in the UPnP application advertisement.		
Handle Blocking	Flag, whether the application will take care of the blocking, in case the MirrorLink Client blocks the content.	bool	Write

1 2

3

5

6

7 8

9

10

11

12

13

14

15

19 20

21

22

23

The MirrorLink Server is responsible for mixing the different audio streams, i.e. application audio as well as system audio, into a single audio stream for the MirrorLink Client. The provided audio context information is attached from the MirrorLink Server to the audio packets, prior sending them out to the MirrorLink Client.

The Audio Context information is used from the MirrorLink Client to mix the received MirrorLink Server audio stream with the internal MirrorLink Client audio. Therefore the audio context information MUST be timely synchronized with the actual audio content. Based on the received context information, the MirrorLink Client has the following basic mixing options:

- 1. The received MirrorLink audio is **blocked**. An audio blocking message will be sent from the MirrorLink Client (see following API call).
- 2. The received MirrorLink audio is **mixed** with the local MirrorLink Client audio. MirrorLink audio goes either into the foreground or into the background. Local audio continues, alone after MirrorLink audio finished.
- 3. The received MirrorLink audio **replaces** the local MirrorLink Client audio. Local MirrorLink audio MAY pause or stop and later resume or restart once the MirrorLink audio finishes.

The provided audio context information is for audio purpose only, and does not necessarily need to classify the application as such, i.e. the audio context information may differ from the provided framebuffer context information.

#### 4.9.4 0x0804 – Audio Blocking Information Callback

Description: Audio is blocked from the MirrorLink Client; in case the application has indicated that it will handle the blocking (refer to 4.9.3) it MUST remove the blocked content.

The MirrorLink Server will handle the Audio Blocking, if the application is unable to handle the callback in time. This MAY include terminating the application.

24 Obligation: Mandatory25 Type: Callback

Feature Name	Description	Type	Direction
Blocking Rea- son	Reason for blocking	uint16	Read

- 1 MirrorLink specified a set of audio blocking reasons, which are provided from the MirrorLink Client. Some
- 2 of these audio blocking notifications will be handled directly from the MirrorLink Server itself, without no-
- 3 tifying the application, whereas others are provided to the applications for their further handling.
- 4 The following overview lists the blocking reasons specified by MirrorLink and how they are handled from
- 5 the MirrorLink Server. The MirrorLink Server SHOULD only pass the notification to the application, if no
- 6 reason flag is set to handle the blocking by itself.

7

8

9

10

11

15

16 17

18

19

2021

22

27

#### • Bit 0 – Not allowed application category

The MirrorLink Server MUST send a callback, if the application will has previously set the audio context information via the Common API function 0x0803 with the "Handle Blocking" parameter set to TRUE. Otherwise the MirrorLink Server does not know, whether the application can handle the blocking and hence no notification will be sent.

The MirrorLink Server MUST continue sending the callback for a limited time, in case of a CCC certified application. Otherwise the MirrorLink Server MUST handle the situation and no further blocking message is send.

#### • Bit 1 – Not sufficient application trust level

MirrorLink Server MUST handle the blocking and no audio blocking notification is sent.

#### • Bit 2 – Not allowed application ID

MirrorLink Server MUST handle the blocking and no audio blocking notification is sent.

#### • Bit 3 – Global audio muted

The MirrorLink Server MUST pass the notification to the application.

#### • Bit 4 – Audio stream, as given by application ID, muted

The MirrorLink Server MUST pass the notification to the application.

- 23 A MirrorLink Server, handling an audio blocking notification MUST either filter the application's audio, or
- 24 terminate the application. The MirrorLink Server MAY terminate an application, providing an audio stream,
- 25 which is getting blocked from the MirrorLink Client. In particular if they continue the audio streaming, even
- being notified to stop the audio streaming.

#### 4.9.5 0x0805 – Framebuffer Unblocking Callback

- 28 Description: Framebuffer is unblocked from the MirrorLink Client. This signal will be emitted, if the
- 29 MirrorLink Server has previously blocked part of the framebuffer using the API call
- 30 0x0802 or via internal handling.
- 31 If multiple bits have been enabled, a framebuffer unblocking callback MUST be only
- 32 called, when all conditions have been met.
- 33 Obligation: Mandatory
- 34 Type: Callback
- 35 Feature List: None
- 36 The following overview lists the reasons of the initial framebuffer blocking, specified by MirrorLink, and
- 37 how they are handled from the MirrorLink Server with respect to the unblocking callback. The MirrorLink
- 38 Server SHOULD only pass the notification to the application, if no reason flag is set to handle the unblocking
- 39 by itself.

#### 40• Bit 0 – Not allowed content category

- 41 MirrorLink Server MUST pass the unblocking notification to the application, as soon as the MirrorLink
- 42 Server receives two consecutive Framebuffer Update Request messages with no Framebuffer Blocking No-
- 43 tification in between AND the application has set the "Handle Blocking" parameter to TRUE in the
- 44 Common API function 0x0801.

#### 45• Bit 1 − Not allowed application category

46 Same behavior as with Bit 0

#### 1 • Bit 2 – Not sufficient content trust level

2 No framebuffer unblocking notification send.

#### 3• Bit 3 – Not sufficient application trust level

4 No framebuffer unblocking notification send.

#### 5• Bit 4 – Content rules not followed

6 No framebuffer unblocking notification send.

#### 7• Bit 5 − Not allowed application ID

8 No framebuffer unblocking notification send.

#### 9• Bit 8 – UI not in focus on remote display

- 10 MirrorLink Server MUST pass the unblocking notification to the application, as soon as the MirrorLink
- 11 Server receives two consecutive Framebuffer Update Request messages with a Framebuffer Blocking Noti-
- 12 fication in between.

13

14

15

16 17

18

30

31

32

33

34

35

3637

38 39

40

41 42

#### • Bit 9 – UI not visible on remote display

MirrorLink Server MUST pass the unblocking notification to the application, as soon as the MirrorLink Server resumes the Framebuffer Updates.

#### • Bit 10 – UI layout not supported (after a Desktop Size Pseudo Encoding)

No framebuffer unblocking notification send.

#### 4.9.6 0x0806 – Audio Unblocking Callback

- 19 Description: Audio is unblocked from the MirrorLink Client. This signal will be emitted, if the Mir-
- 20 rorLink Client has previously blocked application's audio stream. The application will
- 21 receive this signal, as soon as the MirrorLink Client resumes the audio.
- 22 If multiple bits have been enabled, an audio unblocking callback is only called, when all
- conditions have been met.
- 24 Obligation: Mandatory
- 25 Type: Callback
- 26 Feature List: None
- 27 The following overview lists the blocking reasons specified by MirrorLink and how they are handled from
- 28 the MirrorLink Server. The MirrorLink Server SHOULD only pass the notification to the application, if no
- reason flag is set to handle the unblocking by itself.

#### • Bit 0 – Not allowed application category

MirrorLink Server MUST pass the unblocking notification to the application, as soon the MirrorLink Server receives an audio unblocking notification from the MirrorLink Client for the given application ID AND the application has set the "Handle Blocking" parameter to TRUE in the Common API function 0x0803.

#### • Bit 1 – Not sufficient application trust level

No audio unblocking notification send.

#### • Bit 2 – Not allowed application ID

No audio unblocking notification send.

#### • Bit 3 – Global audio muted

MirrorLink Server MUST pass the unblocking notification to the application, as soon as the MirrorLink Server receives an audio unblocking notification from the MirrorLink Client.

#### • Bit 4 – Audio stream, as given by application ID, muted

MirrorLink Server MUST pass the unblocking notification to the application, as soon as the MirrorLink Server receives an audio unblocking notification from the MirrorLink Client for the given application ID.

# 4.10 0x09xx - Device Status Information

#### 2 4.10.1 0x0901 – Drive Mode

3 Description: Check the drive mode status on the MirrorLink Server; requires established VNC connec-

tion

5 Obligation: Mandatory

6 Type: Get

7 Feature List:

4

Feature Name	Description	Type	Direction
Drive Mode	Flag enabling drive mode for the application	bool	Read

#### 8 4.10.2 0x0902 – Drive Mode Callback

9 Description: Enable drive mode on the MirrorLink Server application; requires established VNC con-

10 nection

11 Obligation: Mandatory

12 Type: Callback

13 Feature List:

Feature Name	Description	Type	Direction
Drive Mode	Flag enabling drive mode for the application	bool	Read

#### 14 4.10.3 0x0903 – Night Mode

15 Description: Check the night mode on the MirrorLink Server; requires established VNC connection

16 Obligation: Mandatory

17 Type: Get

18 Feature List:

Feature Name	Description	Туре	Direction
Night Mode	Flag enabling night mode for the application	bool	Read

## 19 4.10.4 0x0904 – Night Mode Callback

20 Description: Enable night mode on the MirrorLink Server application; requires established VNC con-

21 nection

22 Obligation: Mandatory

23 Type: Callback

24 Feature List:

<b>Feature Name</b>	Description	Type	Direction
Night Mode	Flag enabling night mode for the application	bool	Read

#### 25 4.10.5 0x0905 – Microphone State

26 Description: Check the status of the Microphone from the MirrorLink Client; requires established VNC

27 connection

28 Obligation: Conditional – Voice Control or Phone Audio supported over RTP

- 1 Type: Get
- 2 Feature List:

<b>Feature Name</b>	Description	Type	Direction
Mic Input	Flag whether mic input is enabled on MirrorLink Client	bool	Read
Voice Input	Flag whether voice input is enabled	bool	Read

## 3 4.10.6 0x0906 – Open Microphone Callback

4 Description: Response on opening the Microphone from the MirrorLink Client; requires established

5 VNC connection

6 Obligation: Conditional – Voice Control or Phone Audio supported over RTP

7 Type: Callback

8 Feature List:

Feature Name	Description	Type	Direction
Mic Input	Flag whether mic input is enabled on MirrorLink Client	bool	Read
Voice Input	Flag whether voice input is enabled	bool	Read

## 9 4.10.7 0x0907 - Set Open Microphone

10 Description: Open the Microphone on the MirrorLink Client; requires established VNC connection

11 Obligation: Conditional – Voice Control or Phone Audio supported over RTP

12 Type: Set

Feature Name	Description	Туре	Direction
Mic Input	Flag enabling mic input on the MirrorLink Client.	bool	Write
	Flag enabling voice input on the MirrorLink Client	bool	Write
Voice Input	The application MUST set the Mic Input flag to TRUE, if the Voice input flag is set to TRUE.		

#### 4.11 0x0Axx – Data Services

- 2 Theses API functions provide access to Data Services provided from the MirrorLink Client. The APIs cannot
- 3 be used to implement a data service provided from the MirrorLink Server.

#### 4.11.1 0x0A01 – Get Available Services

5 Description: Retrieve list of available Services provided from the MirrorLink Client and supported

from the MirrorLink Server; requires established CDB connection; any later change to the

provided information MUST be notified via the callback function defined in 4.11.2.

The MirrorLink Server will need to check for the application's certification type and the information regarding service certification (using serviceList element in A\_ARG\_TYPE\_AppCertificateInfo) before returning the list of services to the applica-

tion, i.e. an application may not have access to a particular data service, if the MirrorLink

Client has limited access to only specific certified applications.

13 Obligation: Conditional – Data Services Module available

14 Type: Get

15 Feature List:

1

4

7 8

9

10

11

12

Feature Name	Description	Type	Direction
Services Pro-	List of provided services; an empty array is returned if the CDB connection has not been established.	ServiceInfo[]	Read
vided	Default: Empty array		

#### 16 4.11.2 0x0A02 – Available Services Callback

17 Description: Change in available services. Callback must be called, when CDB connection is estab-

18 lished.

19 Obligation: Conditional – Data Services Module available

20 Type: Callback

21 Feature List:

Feature Name	Description	Type	Direction
Services Pro- vided	List of provided services; an empty array is returned if the CDB connection has not been established.  Default: Empty array	ServiceInfo[]	Read

#### 22 4.11.3 0x0A03 – Register to a Service

23 Description: Register to an available Service; requires established CDB connection; asynchronous re-

sponse is provided by the callback specified in 4.11.4.

25 Obligation: Conditional – Data Services Module available

26 Type: Set

27 Feature List:

Feature Name	Description	Type	Direction
Service ID	Service identifier	uint16	Write

#### 28 4.11.4 0x0A04 – Register to a Service Callback

29 Description: Registration completed; asynchronous response to the function specified in section 0.

1 Obligation: Conditional – Data Services Module available

2 Type: Callback

3 Feature List:

Feature Name	Description	Type	Direction
Service ID	Service identifier	uint16	Read
Success	Flag, to indicate whether the action is successful	bool	Read

#### 4 4.11.5 0x0A05 – Unregister from a Service

5 Description: Unregister from an available Service; requires established CDB connection;

6 Obligation: Conditional – Data Services Module available

7 Type: Set

8 Feature List:

Feature Name	Description	Type	Direction
Service ID	Service identifier	uint16	Write

## 9 4.11.6 0x0A06 – Subscribe to an Object

10 Description: Subscribe a Service Object; requires established CDB connection; asynchronous response

is provided by the callback specified in 4.11.7.

12 Obligation: Conditional – Data Services Module available

13 Type: Set

14 Feature List:

Feature Name	Description	Type	Direction
Service ID	Service identifier	uint16	Write
Object ID	Hash value of the object	uint32	Write

# 15 4.11.7 0x0A07 – Subscribe to an Object Callback

16 Description: Subscription complete; asynchronous response to the function specified in 4.11.6. Any

update to the value of the data object will be provided via the Get Object Callback, spec-

18 ified in 4.11.12.

19 Obligation: Conditional – Data Services Module available

20 Type: Callback

21 Feature List:

Feature Name	Description	Type	Direction
Service ID	Service identifier	uint16	Read
Object ID	Hash value of the object	uint32	Read
Success	Flag, to indicate whether the action is successful	bool	Read
Subscription	Subscription type  0x00: Regular interval	uint8	Read
type	0x01: On Change		
	0x02: Automatic		

Feature Name	Description	Type	Direction
	Regular time interval in ms, in which updates are sent.	uint32	Read
Interval	MUST be 0 for subscription types 0x01 (on change) and 0x02 (Automatic).		

#### 1 4.11.8 0x0A08 – Unsubscribe from an Object

2 Description: Unsubscribe from a Service Object

3 Obligation: Conditional – Data Services Module available

4 Type: Set

5 Feature List:

Feature Name	Description	Type	Direction
Service ID	Service identifier	uint16	Write
Object ID	Hash value of the object	uint32	Write

## 6 4.11.9 0x0A09 - Set an Object

7 Description: Set a Service Object; requires established CDB connection and registered service; asyn-

chronous response is provided by the callback specified in 4.11.10

9 Obligation: Conditional – Data Services Module available

10 Type: Set

11 Feature List:

8

Feature Name	Description	Type	Direction
Service ID	Service identifier	uint16	Write
Object ID	Hash value of the object	uint32	Write
Object Value	Pointer to the object's value	void*	Write

## 12 4.11.10 0x0A0A - Set Object Callback

13 Description: Set a Service object completed; requires established CDB connection, asynchronous re-

sponse to the function specified in 4.11.9.

15 Obligation: Conditional – Data Services Module available

16 Type: Callback

17 Feature List:

Feature Name	Description	Type	Direction
Service ID	Service identifier	uint16	Read
Object ID	Hash value of the object	uint32	Read
Success	Flag, to indicate whether the action is successful	bool	Read

#### 18 4.11.11 0x0A0B – Get an Object

19 Description: Get a Service Object; requires established CDB connection and registered service; asyn-

20 chronous response is provided by the callback specified in 4.11.12.

21 Obligation: Conditional – Data Services Module available

1 Type: Get

#### 2 Feature List:

Feature Name	Description	Туре	Direction
Service ID	Service identifier	uint16	Write
Object ID	Hash value of the object	uint32	Write
Object Value	Pointer to the object's value	void*	Read

## 4.11.12 0x0A0C - Get Object Callback

Description: New data object available; requires established CDB connection, registered service and

an object subscription; asynchronous response to the functions specified in 4.11.11. This

callback will be used from the MirrorLink Server to provide new data value for objects to

which the application has subscribed using 4.11.6.

8 Obligation: Conditional – Data Services Module available

9 Type: Callback

10 Feature List:

3 4

5

Feature Name	Description	Туре	Direction
Service ID	Service identifier	uint16	Read
Object ID	Hash value of the object	uint32	Read
Success	Flag, to indicate whether the action is successful	bool	Read
Object Value	Pointer to the object's value	void*	Read

#### 4.12 0x0Bxx - Notifications

#### 2 4.12.1 0x0B01 – Notifications Supported

3 Description: Indicate support for UPnP notifications from the application; the MirrorLink Server will

 $issue\ a\ {\tt NotiAppListUpdate}\ event,\ to\ inform\ the\ MirrorLink\ Client\ that\ the\ notifica-$ 

tion support for this application has changed. Unless otherwise set by the application, the

MirrorLink Server MUST assume that the application will not support notifications.

7 Obligation: Conditional – Notifications Module available

8 Type: Set

9 Feature List:

1

4

5

6

<b>Feature Name</b>	Description	Type	Direction
Notifications supported	Flag indicating notification support from the application	bool	Write

#### 10 4.12.2 0x0B02 – Notifications Enabled

11 Description: Checks whether notifications are enabled for the application; any later change to the pro-

vided information MUST be notified via the callback function defined in 4.12.3.

13 Obligation: Conditional – Notifications Module available

14 Type: Get

15 Feature List:

Feature Name	Description	Type	Direction
Notifications enabled	Flag indicating that notifications are enabled from MirrorLink Server and Client for the application  Default: False	bool	Read

#### 16 4.12.3 0x0B03 – Notifications Enabled Callback

17 Description: Notification enablement has changed.

18 Obligation: Conditional – Notifications Module available

19 Type: Callback

20 Feature List:

Feature Name	Description	Type	Direction
Notifications enabled	Flag indicating that notifications are enabled from MirrorLink Server and Client for the application  Default: False	bool	Read

#### 21 4.12.4 0x0B04 – Notification Configuration

22 Description: Get configuration information for the notification service; any later change to the provided

information MUST be notified via the callback function defined in 4.12.5.

24 Obligation: Conditional – Notifications Module available

25 Type: Get

26 Feature List:

Feature Name	Description	Type	Direction
Notification UI Support	Flag, whether the MirrorLink client supports its own notification UI	bool	Read
Max Actions	Maximum number of actions	uint8	Read
Max Action Name Length	Maximum number of characters of the Action Name	uint8	Read
Max Notifica- tion Title Length	Maximum number of characters of the notification title	uint16	Read
Max Body Length	Maximum number of characters of the notification body.	uint16	Read

# 1 4.12.5 0x0B05 – Notification Configuration Callback

2 Description: Notification Configuration information has changed.

3 Obligation: Conditional – Notifications Module available

4 Type: Callback

5 Feature List:

Feature Name	Description	Туре	Direction
Notification UI Support	Flag, whether the MirrorLink client supports its own notification UI	bool	Read
Max Actions	Maximum number of actions	uint8	Read
Max Action Name Length	Maximum number of characters of the Action Name	uint8	Read
Max Notifica- tion Title Length	Maximum number of characters of the notification title	uint16	Read
Max Body Length	Maximum number of characters of the notification body.	uint16	Read

# 6 4.12.6 0x0B06 - Send Notification for client-based Notification UI

7 Description: Send a notification from the application; this notification replaces a previously send noti-

8 fication.

9 Obligation: Conditional – Notifications Module available

10 Type: Set

Feature Name	Description	Type	Direction
notiTitle	Title of the notification event	string8	Write
notiBody	Body of the notification event	string8	Write
iconUrl	Url to icon belonging to the notification  Icon MUST be of mimetype "image/png" with a color depth of 24.	url	Write

<b>Feature Name</b>	Description	Type	Direction
actionList	List of actions belonging to the notification	Action[]	Write
notificationID	Returns the notification identifier; a Zero value will be returned, if the action was not successful.	uint32	Read

#### 4.12.7 0x0B07 – Send Notification for VNC-based Notification UI

2 Description: Send a notification from the application; this notification replaces a previously send noti-

fication.

4 Obligation: Conditional – Notifications Module available

5 Type: Set

6 Feature List:

3

<b>Feature Name</b>	Description	Type	Direction
notificationID	Returns the notification identifier; a Zero value will be returned, if the action was not successful.	uint32	Read

#### 7 4.12.8 0x0B08 – Cancel Notification

8 Description: Cancel a notification from the application;

9 Obligation: Conditional – Notifications Module available

10 Type: Set

11 Feature List:

Feature Name	Description	Type	Direction
notification ID	Identifier of the notification, which needs to get canceled.	uint32	Write

#### 12 4.12.9 0x0B09 – Receive Action Callback

13 Description: Receive action from the MirrorLink Client for a notification;

14 Obligation: Conditional – Notifications Module available

15 Type: Callback

16 Feature List:

<b>Feature Name</b>	Description	Type	Direction
notification ID	Identifier of the notification	uint32	Read
actionID	Action identifier	uint32	Read

# 4.13 0x0Cxx - Web Application specific Methods

2 Reserved for future use.



## 4.14 0x0Dxx - Misc. MirrorLink 1.2 Additions

#### 2 4.14.1 0x0D01 – MirrorLink Client Driver Distraction Information

3 Description: Provided driver distraction regulation support information of MirrorLink Client, as re-

ceived through the UPnP Client Profile Service; any later change to the provided infor-

mation MUST be notified via the callback function defined in 4.14.2.

6 Obligation: Optional

7 Type: Get

8 Feature List:

1

4

5

Feature Name	Description	Type	Direction
Driver Distrac- tion Support	Flag, to indicate whether the ML Client device supports driver distraction regulation.	Bool	Read
Success	Flag, to indicate whether the information is available	Bool	Read

#### 9 4.14.2 0x0D02 – MirrorLink Client Driver Distraction Callback

10 Description: Indicates that the Client Driver Distraction information has changed.

11 Obligation: Optional

12 Type: Callback

13 Feature List:

Feature Name	Description	Type	Direction
Driver Distrac- tion Support	Indicator whether the ML Client device supports driver distraction regulation.	Bool	Read

# 5 REFERENCES

1

4

5

6

2	[1]	IETF, RFC 2119, "Keys words for use in RFCs to Indicate Requirement Levels", March 1997
3		http://www.ietf.org/rfc/rfc2119.txt

- [2] Car Connectivity Consortium, "MirrorLink Application Server Service", Version 1.1; CCC-TS-024
- [3] Car Connectivity Consortium, "MirrorLink VNC based Display and Control", Version 1.1, CCC-TS-010