Car Connectivity Consortium

MirrorLink®

Common API

Version 1.1.7 & 1.2.2
(CCC-TS-038)

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

Brakensiek, Jörg (Editor)	Microsoft Corporation
Soundararajan, Murali	Samsung
Lünnemann, Patrick	Carmeq GmbH/Volkswagen AG

LEGAL NOTICE

TABLE OF CONTENTS

TERMS AND ABBREVIATIONS

ACMS	Application Certification Management System
BT	Bluetooth
ML	MirrorLink
OCSP	Online Certificate Status Protocol
RFB	Remote Framebuffer
UPnP	Universal Plug and Play
USB	Universal Serial Bus
VNC	Virtual Network Computing

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

1 ABOUT

This document specifies the features of the MirrorLink Common API, available for all MirrorLink Certified Applications on a MirrorLink Certified Server device.

The specification lists a series of requirements, either explicitly or within the text, which are mandatory elements for a compliant solutions. Recommendations are given, to ensure optimal usage and to provide suitable performance. All recommendations are optional.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are following the notation as described in RFC 2119 [1].

1. MUST: This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.
2. MUST NOT: This phrase, or the phrase "SHALL NOT", mean that the definition is an absolute prohibition of the specification.
3. SHOULD: This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
4. SHOULD NOT: This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
5. MAY: This word, or the adjective "OPTIONAL", means that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option MUST be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option MUST be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides.)

2 INTRODUCTION

The Common API specifies an interface to the MirrorLink Server, which allows any application to either get information about MirrorLink Server's or Client's properties or to set them to specific values. In addition, the API specifies callback functions, which are used from the MirrorLink Server to inform the application about any change. Callback functions MUST be implemented from the applications for any evented function.

The Common API specifies the interface in a platform/OS independent manner. Platform specific specification will describe the detailed platform specific view of the Common API, which MUST be implemented 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.

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).

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.

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 platform specific implementation of the Common API MAY use existing platform APIs are sub-classed versions of them.

Data Type	Description
bool	Data type representing the logical values true and false. 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 *typeName*, 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

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

If the Common API replicates functionality, available via OS/Platform APIs, then those API MUST be used, as defined in the Platform specific specifications.

The platform specific API MAY rearrange the defined parameter, or add additional parameter. The platform specific API MUST NOT remove any parameter.

3 DEFINITIONS

3.1 0xE001 – Structure Rect

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

Table 2: Structure Rect

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

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.	uint16
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

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

Table 5: Structure FbContext

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

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 callback functions.

Any MirrorLink Server MUST implement all functions within an optional module, if it supports that module. 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 the operation of the application; the platform specific specification MAY provide conditions for the obligation of individual callback functions.

The MirrorLink applications MUST use the 0x0301 Common API Call and the 0x0302 Common API 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.

MirrorLink Servers MUST have the Common API modules available at all times.

4.1 0xF0xx – MirrorLink Common API Info

4.1.1 0xF001 – MirrorLink Common API Version

Description:	Implemented MirrorLink Common API Version from the MirrorLink Server
Obligation:	Mandatory
Type:	Get

Feature List:

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

4.1.2 0xF002 – Common API Module Available

Description:	Check, whether MirrorLink Server supports a specific Common API module
Obligation:	Mandatory
Type:	Get

Feature List:

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

Description: Available MirrorLink Version for the established connection, as agreed between the MirrorLink 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.

Obligation: Mandatory

Type: Get

Feature List:

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

4.2.2 0x0102 – MirrorLink Version Callback

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

Obligation: Mandatory

Type: Callback

Feature List:

Feature Name	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

4.2.3 0x0103 – MirrorLink Client Manufacturer and Model Information

Description: Provided MirrorLink client manufacturer and model information, as received through the UPnP Client Profile Service; any later change to the provided information MUST be notified via the callback function defined in 4.2.4.

Obligation: Mandatory

Type: Get

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

4.2.4 0x0104 – MirrorLink Client Manufacturer and Model Information Callback

Description: Indicates that the Client information has changed.

Obligation: Mandatory

Type: Callback

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.

Obligation: Mandatory

Type: Get

Feature List:

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 drive-certification	bool	Read

4.3 0x02xx – Certification Information

4.3.1 0x0201 – Get Application Certification Status

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

Obligation: Mandatory

Type: Get

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

4.3.2 0x0202 – Get Application Certifying Entities

Description: Provide information on the certifying entities

Obligation: Mandatory

Type: Get

Feature List:

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

4.3.3 0x0203 – Get Application Certification Information

Description: Provided application certificate information; any later change to the provided information MUST be notified via the callback function defined in 4.3.4,

Obligation: Mandatory

Type: Get

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 (base-level) from the given entity	string8	Read

4.3.4 0x0204 – Get Application Certification Information Callback

Description: Indicate that the application certificate information changed.

Obligation: Mandatory

Type: Callback

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 (base-level) from the given entity	string8	Read

4.4 0x03xx – Connection Information

4.4.1 0x0301 – Established MirrorLink Connection

Description:	Established MirrorLink connection; any later change to the provided information MUST be notified via the callback function defined in 4.4.2.
Obligation:	Mandatory
Type:	Get

Feature List:

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

4.4.2 0x0302 – Established MirrorLink Connection Callback

Description: Indicate that the MirrorLink connection status changed. The callback MUST be provided 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.

A MirrorLink connection is *established* latest in the following situation (whatever comes first):

- MirrorLink Client sends a UPnP `SetClientProfile` action with an non-empty 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)

Obligation: Mandatory

Type: Callback

Feature List:

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

4.4.3 0x0303 – Established Audio Connections

Description:	Established Audio connections within MirrorLink setup; any later change to the provided information MUST be notified via the callback function defined in 4.4.4.
Obligation:	Mandatory
Type:	Get

Feature List:

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

Feature Name	Description	Type	Direction
	0x01: Not established 0x02: BT A2DP 0x03: RTP		
Media Audio In	Identifier of the audio connection for media audio (in-put) 0x00: Not available 0x01: Not established 0x03: RTP	uint8	Read
Voice Control	Identifier of the audio connection for Voice Control au-dio (input) 0x00: Not available 0x01: Not established 0x02: BT HFP + BVRA (Voice Control is outside Mir-rorLink Server's responsibility; application must use existing platform APIs) 0x03: RTP	uint8	Read
Phone Audio	Identifier of the audio connection for Phone audio (in-put & 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

4.4.4 0x0304 – Established Audio Connections Callback

Description: Indicate that the audio connections changed.

Obligation: Mandatory

Type: Callback

Feature List:

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

4.4.5 0x0305 – Established Remote Display Connection

Description: Established remote display connection; any later change to the provided information MUST be notified via the callback function defined in 4.4.6.

Obligation: Mandatory

Type: Get

Feature List:

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

4.4.6 0x0306 – Established Remote Display Connection Callback

Description: Indicate that the remote display connections changed.

Obligation: Mandatory

Type: Callback

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)[1]
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.

Obligation: Mandatory

Type: Get

Feature List:

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

[1] 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 Columns	Number of pad columns added from the MirrorLink Server to the scaled application framebuffer	uint16	Read
Client Horizontal 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
Success	Flag, to indicate whether the information is available	bool	Read

4.5.2 0x0402 – Display Configuration Callback

Description: Display Configuration has changed.

Obligation: Mandatory

Type: Callback

Feature List:

Feature Name	Description	Type	Direction
App Horizontal Resolution	Horizontal resolution in pixel of the framebuffer, the application is rendering into. Note: In many cases, the App Horizontal Resolution equals the horizontal resolution of the MirrorLink Server's display.	uint16	Read
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 Horizontal 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 Columns	Number of pad columns added from the MirrorLink Server to the scaled application framebuffer	uint16	Read
Client Horizontal 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

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

Obligation: Mandatory

Type: Get

Feature List:

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

4.5.4 0x0404 – Client Pixel Format Callback

Description: Pixel format has changed.

Obligation: Mandatory

Type: Callback

Feature List:

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

4.5.5 0x0405 – Set Framebuffer Orientation Support

Description: Inform the MirrorLink Server about the application's framebuffer orientation support; unless otherwise set by the application, the VNC Server MUST assume that the application will only support Landscape.

Obligation: Mandatory

Type: Set

Feature List:

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

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 Callback as specified in 4.5.6.

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.

Obligation: Mandatory

Type: Callback

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 VNC connection; any later change to the provided information MUST be notified via the callback function defined in 4.6.2.

Obligation: Mandatory

Type: Get

Feature List:

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 Support	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

4.6.2 0x0502 – Event Configuration Callback

Description: Client event configuration information has changed.

Obligation: Mandatory

Type: Callback

Feature List:

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 Support	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 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
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

Type: Get

Feature List:

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

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

Obligation: Mandatory

Type: Get

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

4.6.5 0x0505 – Get Event Mapping Callback

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

Obligation: Mandatory

Type: Callback

Feature List:

Feature Name	Description	Type	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

4.7.1 0x0601 – Show Client Virtual Keyboard

Description: Trigger a virtual keyboard at the MirrorLink Client; requires an established VNC connection

Obligation: Conditional – Virtual Keyboard Module available

Type: Set

Feature List:

Feature Name	Description	Type	Direction
Virtual Keyboard 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

4.7.2 0x0602 – Client Virtual Keyboard Support

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

Obligation: Conditional – Virtual Keyboard Module available

Type: Get

Feature List:

Feature Name	Description	Type	Direction
Virtual Keyboard 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

4.7.3 0x0603 – Client Virtual Keyboard Text Entry Callback

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

Obligation: Conditional – Application uses Virtual Keyboard Module

Type: Callback

Feature List:

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

4.8 0x07xx – Key Event Listing

4.8.1 0x0701 – Key Event List

Description:	Provide a white list of key events; key events are following the MirrorLink client device language setting; requires established VNC connection
Obligation:	Conditional – Key Event Listing Module available
Type:	Set
Feature List:	

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

4.8.2 0x0702 – Key Event List Support

Description:	Check, whether MirrorLink client and server support key event listing; requires established VNC connection
Obligation:	Conditional – Key Event Listing Module available
Type:	Get
Feature List:	

Feature Name	Description	Type	Direction
Key Event Listing 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

Description: Provides information of the current framebuffer context; the MirrorLink 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.

Obligation: Mandatory

Type: Set

Feature List:

Feature Name	Description	Type	Direction
Framebuffer Context Information	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

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.

Obligation: Mandatory

Type: Callback

Feature List:

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

MirrorLink specified a set of framebuffer blocking reasons, which are provided from the MirrorLink Client. Some of these framebuffer blocking notifications will be handled directly from the MirrorLink Server itself, 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
- **Bit 2 – Not sufficient content trust level**
  MirrorLink Server MUST handle the blocking and no framebuffer blocking notification is sent.[2]
- **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.[3]
- **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)**

MirrorLink Server MUST handle the blocking and no framebuffer blocking notification is sent.[4]

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.

Obligation: Mandatory

Type: Set

Feature List:

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

[2] The application is a non-certified application.

[3] The MirrorLink Client uses this reason flag, if it blocks an application for certification status reason.

[4] The MirrorLink Server SHOULD use the Switch Framebuffer Orientation Callback (0x0406).

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

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.
Obligation:	Mandatory
Type:	Callback

Feature List:

Feature Name	Description	Type	Direction
Blocking Reason	Reason for blocking	uint16	Read

MirrorLink specified a set of audio blocking reasons, which are provided from the MirrorLink Client. Some of these audio blocking notifications will be handled directly from the MirrorLink Server itself, 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 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.

A MirrorLink Server, handling an audio blocking notification MUST either filter the application's audio, or terminate the application. The MirrorLink Server MAY terminate an application, providing an audio stream, 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

Description:	Framebuffer is unblocked from the MirrorLink Client. This signal will be emitted, if the MirrorLink Server has previously blocked part of the framebuffer using the API call 0x0802 or via internal handling. If multiple bits have been enabled, a framebuffer unblocking callback MUST be only called, when all conditions have been met.
Obligation:	Mandatory
Type:	Callback
Feature List:	None

The following overview lists the reasons of the initial framebuffer blocking, specified by MirrorLink, and how they are handled from the MirrorLink Server with respect to the unblocking callback. 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 content category**
  MirrorLink Server MUST pass the unblocking notification to the application, as soon as the MirrorLink Server receives two consecutive Framebuffer Update Request messages with no Framebuffer Blocking Notification in between AND the application has set the "`Handle Blocking`" parameter to `TRUE` in the Common API function 0x0801.
- **Bit 1 – Not allowed application category**
  Same behavior as with Bit 0

- **Bit 2 – Not sufficient content trust level**
No framebuffer unblocking notification send.
- **Bit 3 – Not sufficient application trust level**
No framebuffer unblocking notification send.
- **Bit 4 – Content rules not followed**
No framebuffer unblocking notification send.
- **Bit 5 – Not allowed application ID**
No framebuffer unblocking notification send.
- **Bit 8 – UI not in focus on remote display**
MirrorLink Server MUST pass the unblocking notification to the application, as soon as the MirrorLink Server receives two consecutive Framebuffer Update Request messages with a Framebuffer Blocking Notification in between.
- **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

Description:	Audio is unblocked from the MirrorLink Client. This signal will be emitted, if the MirrorLink Client has previously blocked application's audio stream. The application will receive this signal, as soon as the MirrorLink Client resumes the audio. If multiple bits have been enabled, an audio unblocking callback is only called, when all conditions have been met.
Obligation:	Mandatory
Type:	Callback
Feature List:	None

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 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

4.10.1 0x0901 – Drive Mode

Description: Check the drive mode status on the MirrorLink Server; requires established VNC connection

Obligation: Mandatory

Type: Get

Feature List:

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

4.10.2 0x0902 – Drive Mode Callback

Description: Enable drive mode on the MirrorLink Server application; requires established VNC connection

Obligation: Mandatory

Type: Callback

Feature List:

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

4.10.3 0x0903 – Night Mode

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

Obligation: Mandatory

Type: Get

Feature List:

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

4.10.4 0x0904 – Night Mode Callback

Description: Enable night mode on the MirrorLink Server application; requires established VNC connection

Obligation: Mandatory

Type: Callback

Feature List:

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

4.10.5 0x0905 – Microphone State

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

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

Type: Get

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

4.10.6 0x0906 – Open Microphone Callback

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

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

Type: Callback

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

4.10.7 0x0907 – Set Open Microphone

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

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

Type: Set

Feature List:

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

4.11 0x0Axx – Data Services

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

4.11.1 0x0A01 – Get Available Services

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 application, 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.
Obligation:	Conditional – Data Services Module available
Type:	Get

Feature List:

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

4.11.2 0x0A02 – Available Services Callback

Description:	Change in available services. Callback must be called, when CDB connection is established.
Obligation:	Conditional – Data Services Module available
Type:	Callback

Feature List:

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

4.11.3 0x0A03 – Register to a Service

Description:	Register to an available Service; requires established CDB connection; asynchronous response is provided by the callback specified in 4.11.4.
Obligation:	Conditional – Data Services Module available
Type:	Set

Feature List:

Feature Name	Description	Type	Direction
Service ID	Service identifier	uint16	Write

4.11.4 0x0A04 – Register to a Service Callback

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

Obligation: Conditional – Data Services Module available

Type: Callback

Feature List:

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

4.11.5 0x0A05 – Unregister from a Service

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

Obligation: Conditional – Data Services Module available

Type: Set

Feature List:

Feature Name	Description	Type	Direction
Service ID	Service identifier	uint16	Write

4.11.6 0x0A06 – Subscribe to an Object

Description: Subscribe a Service Object; requires established CDB connection; asynchronous response is provided by the callback specified in 4.11.7.

Obligation: Conditional – Data Services Module available

Type: Set

Feature List:

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

4.11.7 0x0A07 – Subscribe to an Object Callback

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, specified in 4.11.12.

Obligation: Conditional – Data Services Module available

Type: Callback

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 type	Subscription type 0x00: Regular interval 0x01: On Change 0x02: Automatic	uint8	Read

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

4.11.8 0x0A08 – Unsubscribe from an Object

Description: Unsubscribe from a Service Object

Obligation: Conditional – Data Services Module available

Type: Set

Feature List:

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

4.11.9 0x0A09 – Set an Object

Description: Set a Service Object; requires established CDB connection and registered service; asynchronous response is provided by the callback specified in 4.11.10

Obligation: Conditional – Data Services Module available

Type: Set

Feature List:

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

4.11.10 0x0A0A – Set Object Callback

Description: Set a Service object completed; requires established CDB connection, asynchronous response to the function specified in 4.11.9.

Obligation: Conditional – Data Services Module available

Type: Callback

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

4.11.11 0x0A0B – Get an Object

Description: Get a Service Object; requires established CDB connection and registered service; asynchronous response is provided by the callback specified in 4.11.12.

Obligation: Conditional – Data Services Module available

Type: Get

Feature List:

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*	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.

Obligation: Conditional – Data Services Module available

Type: Callback

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
Object Value	Pointer to the object's value	void*	Read

4.12 0x0Bxx – Notifications

4.12.1 0x0B01 – Notifications Supported

Description:	Indicate support for UPnP notifications from the application; the MirrorLink Server will issue a `NotiAppListUpdate` event, to inform the MirrorLink Client that the notification support for this application has changed. Unless otherwise set by the application, the MirrorLink Server MUST assume that the application will not support notifications.
Obligation:	Conditional – Notifications Module available
Type:	Set
Feature List:	

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

4.12.2 0x0B02 – Notifications Enabled

Description:	Checks whether notifications are enabled for the application; any later change to the provided information MUST be notified via the callback function defined in 4.12.3.
Obligation:	Conditional – Notifications Module available
Type:	Get
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

4.12.3 0x0B03 – Notifications Enabled Callback

Description:	Notification enablement has changed.
Obligation:	Conditional – Notifications Module available
Type:	Callback
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

4.12.4 0x0B04 – Notification Configuration

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.
Obligation:	Conditional – Notifications Module available
Type:	Get
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 Notification 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

4.12.5 0x0B05 – Notification Configuration Callback

Description: Notification Configuration information has changed.

Obligation: Conditional – Notifications Module available

Type: Callback

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 Notification 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

4.12.6 0x0B06 – Send Notification for client-based Notification UI

Description: Send a notification from the application; this notification replaces a previously send notification.

Obligation: Conditional – Notifications Module available

Type: Set

Feature List:

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

Feature Name	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

Description: Send a notification from the application; this notification replaces a previously send notification.

Obligation: Conditional – Notifications Module available

Type: Set

Feature List:

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

4.12.8 0x0B08 – Cancel Notification

Description: Cancel a notification from the application;

Obligation: Conditional – Notifications Module available

Type: Set

Feature List:

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

4.12.9 0x0B09 – Receive Action Callback

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

Obligation: Conditional – Notifications Module available

Type: Callback

Feature List:

Feature Name	Description	Type	Direction
notification ID	Identifier of the notification	uint32	Read
actionID	Action identifier	uint32	Read

4.13 0x0Cxx – Web Application specific Methods

Reserved for future use.

4.14 0x0Dxx – Misc. MirrorLink 1.2 Additions

4.14.1 0x0D01 – MirrorLink Client Driver Distraction Information

Description: Provided driver distraction regulation support information of MirrorLink Client, as received through the UPnP Client Profile Service; any later change to the provided information MUST be notified via the callback function defined in 4.14.2.

Obligation: Optional

Type: Get

Feature List:

Feature Name	Description	Type	Direction
Driver Distraction 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

4.14.2 0x0D02 – MirrorLink Client Driver Distraction Callback

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

Obligation: Optional

Type: Callback

Feature List:

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

5 References

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

[2] Car Connectivity Consortium, “MirrorLink - 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