Car Connectivity Consortium

MirrorLink®

MirrorLink 1.1 Errata

Version 0.2.1
(CCC-TS-034)

VERSION HISTORY

Version	Date	Comment
0.1.0	29 January 2014	[030-001], [030-002], [030-003] approved
0.1.1	18 March 2014	[010-001], [010-002], [010-003], [010-004], [010-005], [010-006], [010-007], [014-001], [014-002], [020-001], [022-001], [024-001], [024-002], [024-003], [024-004], [030-004], [030-005], [030-006], [030-007] approved
0.1.2	17 June 2014	[010-008], [010-009], [010-010], [010-011], [010-012], [014-003], [024-005], [024-006], [028-001], [028-002]
0.1.3	25 September 2014	[014-014]
0.1.4	10 November 2014	[024-003]: Correct UPnP action name [010-013], [010-014], [016-001], [020-002], [022-002], [024-007], [024-008], [032-001], [036-001], [036-002], [036-003]
0.1.5	18 March 2015	[010-015], [010-016], [010-017], [010-018], [016-002], [012-001], [012-002], [012-003], [024-009], [024-010], [024-011], [024-012], [024-013], [026-001], [026-002], [026-003], [026-004], [030-008], [030-009]
0.2.0	17 June 2015	Integrated the following Errata into the Specification: [010-001], [010-002], [010-003], [010-004], [010-005], [010-006], [010-007], [010-008], [010-009], [010-010], [010-011], [010-012], [010-013], [010-014], [010-015], [010-016], [010-017], [010-018], [036-001], [036-002], [036-003] Added the following item [024-014], [030-010], [036-004], [036-005]
0.2.1	09 September 2015	[012-004]

LIST OF CONTRIBUTORS

Brakensiek, Jörg (Editor) Car Connectivity Consortium LLC

LEGAL NOTICE

The copyright in this Specification is owned by the Car Connectivity Consortium LLC ("CCC LLC"). Use of this Specification and any related intellectual property (collectively, the "Specification"), is governed by these license terms and the CCC LLC Limited Liability Company Agreement (the "Agreement").

Use of the Specification by anyone who is not a member of CCC LLC (each such person or party, a "Member") is prohibited. The legal rights and obligations of each Member are governed by the Agreement and their applicable Membership Agreement, including without limitation those contained in Article 10 of the LLC Agreement.

CCC LLC hereby grants each Member a right to use and to make verbatim copies of the Specification for the purposes of implementing the technologies specified in the Specification to their products ("Implementing Products") under the terms of the Agreement (the "Purpose"). Members are not permitted to make available or distribute this Specification or any copies thereof to non-Members other than to their Affiliates (as defined in the Agreement) and subcontractors but only to the extent that such Affiliates and subcontractors have a need to know for carrying out the Purpose and provided that such Affiliates and subcontractors accept confidentiality obligations similar to those contained in the Agreement. Each Member shall be responsible for the observance and proper performance by such of its Affiliates and subcontractors of the terms and conditions of this Legal Notice and the Agreement. No other license, express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein.

Any use of the Specification not in compliance with the terms of this Legal Notice, the Agreement and Membership Agreement is prohibited and any such prohibited use may result in termination of the applicable Membership Agreement and other liability permitted by the applicable Agreement or by applicable law to CCC LLC or any of its members for patent, copyright and/or trademark infringement.

THE SPECIFICATION IS PROVIDED "AS IS" WITH NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION ANY WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS, AND COMPLIANCE WITH APPLICABLE LAWS.

Each Member hereby acknowledges that its Implementing Products may be subject to various regulatory controls under the laws and regulations of various jurisdictions worldwide. Such laws and regulatory controls may govern, among other things, the combination, operation, use, implementation and distribution of Implementing Products. Examples of such laws and regulatory controls include, but are not limited to, road safety regulations, telecommunications regulations, technology transfer controls and health and safety regulations. Each Member is solely responsible for the compliance by their Implementing Products with any such laws and regulations and for obtaining any and all required authorizations, permits, or licenses for their Implementing Products related to such regulations within the applicable jurisdictions.

Each Member acknowledges that nothing in the Specification provides any information or assistance in connection with securing such compliance, authorizations or licenses**.**

NOTHING IN THE SPECIFICATION CREATES ANY WARRANTIES, EITHER EXPRESS OR IMPLIED, REGARDING SUCH LAWS OR REGULATIONS. ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF ANY INTELLECTUAL PROPERTYRIGHTS OR FOR NONCOMPLIANCE WITH LAWS, RELATING TO USE OF THE SPECIFICATION IS EXPRESSLY DISCLAIMED. BY USE OF THE SPECIFICATION, EACH MEMBER EXPRESSLY WAIVES ANY CLAIM AGAINST CCC LLC AND ITS MEMBERS RELATED TO USE OF THE SPECIFICATION.

CCC LLC reserve the right to adopt any changes or alterations to the Specification as it deems necessary or appropriate.

TABLE OF CONTENTS

VERSION HISTORY **2**
LIST OF CONTRIBUTORS **2**
LEGAL NOTICE **3**
TABLE OF CONTENTS **4**
TERMS AND ABBREVIATIONS **5**
ABOUT **6**
INTRODUCTION **7**
CCC-TS-010: VNC BASED DISPLAY AND CONTROL **8**
CCC-TS-012: AUDIO **9**
CCC-TS-014: DEVICE ATTESTATION PROTOCOL **11**
CCC-TS-016: COMMON DATA BUS **13**
CCC-TS-020: GPS DATA SERVICE **14**
CCC-TS-022: LOCATION DATA SERVICE **15**
CCC-TS-024: UPNP APPLICATION SERVER SERVICE **16**
CCC-TS-026: UPNP CLIENT PROFILE SERVICE **21**
CCC-TS-028: UPNP NOTIFICATION SERVER SERVICE **23**
CCC-TS-030: UPNP SERVER DEVICE **24**
CCC-TS-032: CORE ARCHITECTURE **27**
CCC-TS-036: HANDLING OF APPLICATION CERTIFICATES **28**
REFERENCES **29**

TERMS AND ABBREVIATIONS

BT	Bluetooth
ML	MirrorLink
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.

ABOUT

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

The document will focus on Errata for the MirrorLink 1.1 specification version.

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.)

INTRODUCTION

This document contains all Errata for MirrorLink 1.1 specification. The document distinguishes between the following types of changes

- EDITORIAL Rearrange the words for better readability.
- CLARIFICATION Add additional text to help with understanding the content/context.
- BUG FIX Change text to correct any wrong functional statement in the document.

Note: Grammar and spelling errors as well as any formatting problems will be corrected within the documents directly.

Within the Errata document, orange text highlights text, which SHALL be removed from the original specification document. Blue text highlights text, which SHALL be added to the original specification. A […] indicates that some specification text has been left out for readability purpose. The text left out SHALL NOT be subject to the specific errata change.

Approved Errata Items have a unique identifier, consisting of the document number followed by a "-" and a sequence number. E.g. `[030-002]` identifies the 2nd errata item in the UPnP Server Device specification (CCC-TS-030).

CCC-TS-010: VNC BASED DISPLAY AND CONTROL

The overview of Errata changes are listed below. Those reference the 1.1.7 version of the VNC based Display and Control specification (CCC-TS-010).

- [010-001] – [010-018] have been integrated into version 1.1.7

Identifier	Description	Type	Status
[010-019]			

The individual errata changes are specified below in detail.

CCC-TS-012: AUDIO

The overview of Errata changes are listed below. Those reference the 1.1.3a version of the Audio specification (CCC-TS-012).

Identifier	Description	Type	Status
[012-001]	RTP Audio context	Clarification	Approved, 18.03.2015
[012-002]	No empty RTP packets	Clarification	Approved, 18.03.2015
[012-003]	System Sound handling	Clarification	Approved, 18.03.2015
[012-004]	Empty RTP packets after pause/mute	Clarification	Approved, 09.09.2015

The individual errata changes are specified below in detail.

[012-001] – Chapter 5.1, Page 18, Lines 10-17 – RTP Audio context

CLARIFICATION (2015-03-18) – Clarification is needed to describe how the MirrorLink Server is providing the audio context information, received via the Common API.

The following paragraph is updated:

> The MirrorLink Server MUST use the audio context information, provided from MirrorLink certified and MirrorLink-aware applications via the Common API. The MirrorLink Server MUST use the correct application identifier of the application, using the Common API.
>
> Otherwise for applications not using the Common API, t~~T~~he MirrorLink Server SHOULD determine the source of the application audio streams ~~originating from non-certified MirrorLink applications and from applications not using the Common API **Error! Reference source not found.**~~ or MUST set the application identifier to 0x000000000, and ~~The MirrorLink Server~~ MUST set the audio context information to ~~valid, not unknown values, if it can determine the source of an audio stream. The MirrorLink Server MUST set the audio context information to~~ 0x00000000~~, if it cannot determine the source of an audio stream~~.
>
> RTP packets without a valid RTP Header Extension SHOULD be treated similar to RTP packets received from an unknown application.

[012-002] – Chapter 5.4, Page 22, Line 11 – No empty RTP packets

CLARIFICATION (2015-03-18) – Clarification is needed to prevent the RTP Server sending empty RTP packets or an RTP payload representing a zero volume audio signal.

The following paragraph is updated:

> The MirrorLink Server MUST only send RTP packets, if they carry real audio content. I.e. the RTP Server MUST NOT send empty RTP packets or RTP packets with a zero volume audio signal, if no audio is available. Otherwise the MirrorLink Client is not able to switch back to its entertainment audio source in that situation.

[012-003] – Chapter 5.4, Page 22, Lines 12 – System Sound handling

CLARIFICATION (2015-03-18) – Clarification is needed to describe how the ML Server has to handle system sound.

The following paragraph is updated:

> The MirrorLink Server MUST prevent any system sound being sent to the MirrorLink Client. Possible implementation restrictions are detailed in the platform specific specifications.

[012-004] – Chapter 5.4, Page 22, Line 11 – Empty RTP packets after pause/mute

CLARIFICATION (2015-09-09) – Errata item [012-002] highlighted that the RTP server must not send empty RTP packets or an RTP payload representing a zero volume audio signal. A clarification is needed that after the audio stream has been paused or muted, the Audio server may send empty RTP packets for a short time frame:

The following paragraph is updated:

> The MirrorLink Server MUST only send RTP packets, if they carry real audio content. I.e. the RTP Server MUST NOT send empty RTP packets or RTP packets with a zero volume audio signal, if no audio content is available. Otherwise some MirrorLink Client may not be able to switch back to its entertainment audio source in that situation.
>
> When an audio source has indicated that it been paused or stopped, the RTP Server MUST NOT stream any additional RTP packets with unknown application category or unknown application identifier, from that audio source (i.e. no potential guard interval is streamed and exposed to the RTP Client). The RTP Server MUST set the M-flag to "1" in the last RTP packet from that audio source, if no other audio sources are currently playing; in latter case the RTP extension header MUST NOT include references to the paused/stopped audio source.
>
> If available, the MirrorLink Client MAY switch to local entertainment sources, after the RTP source on the MirrorLink Server has been paused or stopped.
>
> The RTP Client MAY initially receive RTP packets with application identifier and/or application category set to 0x00 for up to 1000ms, in case the MirrorLink application is using an asynchronous mechanism to set the audio context information and to start the audio stream. It is up to the RTP Client to play, to silently ignore or to block those initial RTP packets.

CCC-TS-014: DEVICE ATTESTATION PROTOCOL

The overview of Errata changes are listed below. Those reference the 1.1.6 version of the Device Attestation Protocol specification (CCC-TS-014).

Identifier	Description	Type	Status
[014-001]	Details on DAP failure handling	Clarification	Approved, 18.03.2014
[014-002]	Extended Key Usage	Clarification	Approved, 18.03.2014
[014-003]	Optional applicationPublicKey in XSD	Bug Fix	Approved, 17.06.2014
[014-004]	DAP Response Time	Clarification	Approved, 25.09.2014

The individual errata changes are specified below in detail.

[014-001] – Chapter 4, Page 13, Line 10/12 – Details on DAP failure handling

CLARIFICATION (2014-03-18) – Clarify how to proceed, in case DAP fails. Extend current sentence, with additional detail, on how to proceed the DAP, in case the DAP fails in step (2). Correct text is:

> If the measurement does not matches expected value, the Device Attestation Server sends a DAP `AttestationResponse` with error code 5. If the measurement matches expected value, the ~~attestation protocol~~ Device Attestation Server proceeds with the next step of the attestation protocol as shown in Figure 3.

[014-002] – Chapter 4, Page 12, Line 1-3 – Extend Key Usage

CLARIFICATION (2014-03-18) – Clarify the certificate settings for the DAP Device Certificate. TCG specifies different options for implement. The provided reference may not be specific enough.

> The prerequisite of successful Device Attestation Protocol run is that the MirrorLink server has a X.509 device certificate (with Extended Key Usage tcg-kp-AIKCertificate OID 2.23.133.8.3 as specified in section 3.5 of [14]) for its device key pair from the server device manufacturer. Note: The MirrorLink Client MUST NOT expect other X.509 certificate extensions, mandated e.g. in section 3.4 or 3.5 of [14].

[014-003] – Appendix A, Page 22, Line 27-28 – Optional applicationPublicKey in XSD

BUG FIX (2014-06-17) – The application public key is optional in the attestation response. The XSD schema requires the following bug fix:

```
<xs:element name="applicationPublicKey" type="xs:string"
minOccurs="~~1~~0" maxOccurs="1"/>
```

[014-004] – Chapter 3.1.1, Page 10, Line 11-14 & 23-24– DAP Response Time

CLARIFICATION (2014-09-25) – The DAP specification does not define how long the MirrorLink Server's DAP endpoint must wait for the ML Client endpoint to connect.

The specification is corrected in the following way in Chapter 3.1.1.2:

> Otherwise a new DAP session MUST be established, given the following steps:
>
> 1 DAP server MUST listen for the DAP client to make TCP connection at the provided URL for at least 10s.
>
> 2 DAP client MUST make a TCP connection to the provided URL.

3 DAP server and client MUST start DAP according to the steps defined in Section **Error! Reference source not found.**.

The specification is corrected in the following way in Chapter 3.1.2:

The DAP client MUST wait for any outstanding Device Attestation Response messages, for at least 10s, prior to terminating the DAP session. The DAP Server MUST provide a DAP response to any DAP request within 10s.

Applicable for MirrorLink 1.2 Specification: YES

CCC-TS-016: COMMON DATA BUS

The overview of Errata changes are listed below. Those reference the 1.1.3 version of the Common Data Bus specification (CCC-TS-016).

Identifier	Description	Type	Status
[016-001]	Correct ByeBye sequence	Bug Fix	Approved, 10.11.2014
[016-002]	Correct ServicePayload error response	Bug Fix	Approved, 18.03.2015

The individual errata changes are specified below in detail.

[016-001] – Chapter 4, Page 17, Table 12 – Correct ByeBye Handling

BUG FIX (2014-11-10) – The CDB Server endpoint is required to send its own CDB ByeBye message, upon reception of a CDB ByeBye message from the CDB Client endpoint. The CDB Client endpoint must not send its own CDB ByeBye message, upon reception of a CDB ByeBye message from the CDB Server endpoint. Therefore the correct Table entry is:

Message Received (from)	Message Responses	Comment
ByeBye (~~Client~~Server Endpoint)	No response	Terminate the CDB session
ByeBye (~~Server~~Client Endpoint)	ByeBye	Terminate the CDB session

[016-001] – Chapter 4, Page 17, Table 12 – Correct ServicePayload error response

BUG FIX (2015-03-18) – The CDB endpoint is required to send CDB ServiceReponse message with error code 0x0102 when receiving a ServicePayload message, and the respective service is not running. This is stated wrongly in the current specification.

Therefore the correct Table entry is:

Message Received (from)	Message Responses	Comment
ServicePayload (Sink)	ServiceResponse (0x010~~1~~2)	Payload failed – Service not running
ServicePayload (Source)	ServiceResponse (0x010~~1~~2)	Payload failed – Service not running

CCC-TS-020: GPS DATA SERVICE

The overview of Errata changes are listed below. Those reference the 1.1.1a version of the GPS Data Service specification (CCC-TS-020).

Identifier	Description	Type	Status
[020-001]	Value of timestamp in absence of time	Clarification	Approved, 18.03.2014
[020-002]	Signaling "Not available"	Clarification	Approved, 10.11.2014

The individual errata changes are specified below in detail.

[020-001] – Chapter 3, Page 8, 1st table – Value of timestamp in absence of time

CLARIFICATION (2014-03-18) – Clarify `timeStamp` value, under specific condition. Add the following text:

> In case UTC time associated with the NMEA sentence is unknown, the GPS Data Source MUST set the `timestamp` value to zero (0).

[020-002] – Chapter 4, Page 9, Line 2-6 – Signaling "Not available"

CLARIFICATION (2014-11-10) – The specification is a big vague, when to signal that GPS is not available compared to the GPS receiver is not available (e.g. not functioning correctly). Loss of reception can be encoded into the signal as well, using the respective data objects. Therefore, the following revisions are made:

> Server implementation MUST support at least one active command. The client SHOULD first Get NMEA_description Object, and then it SHOULD Subscribe the NMEA Object.
>
> If the ~~current~~ GPS ~~location~~ receiver is not available, the SBP source endpoint MUST return "Not available" error code. This applies to the case when the GeoLocation Object is ~~S~~subscribed. Then SBP sink endpoint SHOULD send Subscribe command again not earlier than 5s and not later than 30s to get notification again.
>
> If the GPS receiver has only lost satellite reception, it SHOULD encode this into to NMEA sentences and SHOULD NOT return a "Not available" error code.

CCC-TS-022: LOCATION DATA SERVICE

The overview of Errata changes are listed below. Those reference the 1.1.1a version of the Location Data Service specification (CCC-TS-022).

Identifier	Description	Type	Status
[022-001]	Value of timestamp in absence of time	Clarification	Approved, 18.03.2014
[022-002]	Signaling “Not available”	Clarification	Approved, 10.11.2014

The individual errata changes are specified below in detail.

[022-001] – Chapter 3, Page 9 – Value of timestamp in absence of time

CLARIFICATION (2014-03-18) – Clarify `timeStamp` value, under specific condition. Add the following text:

> In case UTC time associated with the acquisition of the location is unknown, the Location Data Source MUST set the `timestamp` value to zero (0).

[022-002] – Chapter 3, Page 9, Line 2-13 – Signaling “Not available”

CLARIFICATION (2014-11-10) – The specification is a big vague, when to signal that GPS is not available compared to the GPS receiver is not available (e.g. not functioning correctly). Loss of reception can be encoded into the signal as well, using the respective data objects.

Therefore, the following revisions are made in Chapter 3:

> GeoLocation Object returns the current geo-location directly. If the ~~current~~ geo-location sensor is not available, server MUST return “Not available” error code. This applies to the case when the GeoLocation Object is s~~S~~ubscribed. ~~When the positioning method is no longer available due to the reason like lost satellite reception, server MUST return “Not available” error code.~~ Then SBP sink endpoint SHOULD send Subscribe command again not earlier than 5s and not later than 30s to get notification again.
>
> When only parts of the location object are not available e.g. due to lost satellite reception, the SBP data source SHOULD set the respective values to NaN.
>
> GeoLocation Object owns Coordinate STRUCTURE as member variable. When a positioning method does not support some member variables like “heading”, server can skip that member variable. But all supported member variables MUST be always present during the CDB session so that client does not need to check if some feature is supported or not again and again. If ~~part of~~ data sensor is temporarily unavailable, the server MUST return “Not available” error code. Default value for all optional member variables ~~are~~ is NaN (Not a Number) as defined in the IEEE standard **Error! Reference source not found.**. Note that default value has meaning when the received data is delivered to upper layer.

CCC-TS-024: UPnP Application Server Service

The overview of Errata changes are listed below. Those reference the 1.1.6 version of the UPnP Application Server Service specification (CCC-TS-024).

Identifier	Description	Type	Status
[024-001]	Change of application meta-information	Clarification	Approved, 18.03.2014
[024-002]	Remove OPTIONAL statement.	Clarification	Approved, 18.03.2014
[024-003]	Use Filter in Home Screen application	Clarification	Approved, 18.03.2014
[024-004]	appCertificateURL	Clarification	Approved, 18.03.2014
[024-005]	allowedProfileIDs default value	Bug fix	Approved, 17.06.2014
[024-006]	allowedProfileIDs default value in XSD	Bug fix	Approved, 17.06.2014
[024-007]	Terminate apps, while switching	Clarification	Approved, 10.11.2014
[024-008]	Launch apps provided via Home Screen	Clarification	Approved, 10.11.2014
[024-009]	Use of Application Certificate Info	Clarification	Approved, 18.03.2015
[024-010]	Return optional element in AppList	Clarification	Approved, 18.03.2015
[024-011]	Definition of restricted/nonRestricted	Clarification	Approved, 18.03.2015
[024-012]	Usage of AppCertFilter	Clarification	Approved, 18.03.2015
[024-013]	List of locales extensible	Clarification	Approved, 18.03.2015
[024-014]	Wildcard in A_ARG_TYPE_AppID	Clarification	Approved, 17.06.2015

The individual errata changes are specified below in detail.

[024-001] – Appendix A, Page 51, Table 6.1 – Change of application meta-information

CLARIFICATION (2014-03-18) – Clarify how the information covered under the application certificate trust level can be altered.

> The user ~~and the application~~ MUST NOT be able to change the values. The application MUST NOT be able to change the values, other than through the Common API.

[024-002] – Chapter 2.2.2, Page 8, Line 26 – Remove OPTIONAL statement

CLARIFICATION (2014-03-18) – The statement about the obligation for the evented variable (OPTIONAL) is misleading, as the obligation for the UPnP Server is mandatory, as shown in Table 2.1. The UPnP specification does not list the obligation for the UPnP Control Point.

> ~~This state variable is OPTIONAL.~~ It is evented, implying that clients can subscribe to receive notifications every time the variable changes using UPnP standardized eventing mechanisms.

[024-003] – Chapter 3.7, Page 37, Line 23-25 – Use Filter in Home Screen application

CLARIFICATION (2014-03-18) – The statement how the MirrorLink Server should use the MirrorLink Client's AppCertFilter entry to determine, whether an application is driver certified for a specific locale is not sufficient.

> In Drive mode, the Home Screen Application ~~MAY~~ MUST consider MirrorLink Client `AppCertFilter` settings, for the `restricted/nonRestricted` entries, to not make an application available, but only based on locale information.

The MirrorLink Server MUST remove applications from the home screen application, or do not allow to launch them from there, during drive mode, when the application is not drive certified in the locales, which are included from the MirrorLink Client in the `restricted` field of the `GetCertifiedApplicationsList` and `GetAppCertificationStatus` action's `AppCertFilter` parameter. Multiple actions MUST be considered as an OR connected filtering.

Note that the MirrorLink Client MAY check for multiple locales.

If the HU does not use the `AppCertFilter`, the MirrorLink Server MUST treat all applications drive certified, which have a non-empty `restricted` entry. In case an application is blocked from the MirrorLink Client during drive mode, the MirrorLink Server MAY remove that application from the Home Screen during drive mode for this MirrorLink session.

[024-004] – Chapter 2.2.6, Page 11, Table 2-3 – appCertificateURL

CLARIFICATION (2014-03-18) – The description of the `appCertificateURL` is meant to be informative, in case the MirrorLink Client likes to check for additional entries, not included in the App Certificate Information. In particular the available information at the provided URL is not be sufficient to validate the certificates trust chain. Trust chain validation is done at the MirrorLink Sever.

URL where application certificate~~:~~ is available. The MirrorLink Client ~~Head Unit~~ uses the certificate ~~to verify trust worthiness of the application information provided~~ for information purpose. The MirrorLink Client MUST NOT validate the certificate's trust chain.

MirrorLink Client MUST use HTTP-GET to access the certificate behind the URL.

[024-005] – Chapter 2.2.6, Page 11, Table 2-3 – allowedProfileIDs default value

BUG FIX (2014-06-17) – The description of the `allowedProfileIDs` states that if the list value is not explicitly specified then it MUST be assumed that the application can be executed using any profile with a valid `profileID`. But the default is stated to be "`0`". Therefore the description of `allowedProfileIDs` entry must be changed.

Comma separated list of client profile identifier (A_ARG_TYPE_ProfileID), for which this application can currently be launched and executed;
If this list is not explicitly specified then it MUST be assumed that the application can be executed using any profile with a valid profileID.
In case the application cannot be launched at a given time via any of the available profiles, then the value of this element MUST be set equal to -1
(A_ARG_TYPE_String)
Default: "~~0~~" (empty string)

[024-006] – Chapter 4.2, Page 41, Line 1-2 – allowedProfileIDs default value in XSD

BUG FIX (2014-06-17) – The default value of the allowedProfileIDs default must be changed to an empty list.

```
<xs:element name="allowedProfileIDs" type="xs:string"
minOccurs="0" default="~~0~~"/>
```

[024-007] – Chapter 2.5.3, Page 22, Line 26-29 – Terminate apps, while switching

CLARIFICATION (2014-11-10) – The obligation language does not contain NOT REQUIRED. From the surrounding text this is a SHOULD NOT requirement. Therefore the text has to be changed to:

The client ~~is~~ SHOULD NOT ~~REQUIRED to~~ call the `TerminateApplication` action every time it switches to a different remote application. If a client launches a different remote application without terminating the existing one, the MirrorLink Server MUST either send the original application to the background or it MUST terminate the original application.

[024-008] – Chapter 3.7, Page 37, Line 31 – Launch apps provided via Home Screen

CLARIFICATION (2014-11-10) – The specification is not very clear, whether or not the MirrorLink Client can use UPnP control actions to e.g. launch an applications, which is included within the MirrorLink Server's home screen applications and therefore flagged with resource status "NA".

Therefore the text has to be extend by:

> The MirrorLink Client MAY use any of the specified UPnP actions to control applications with `protocolID` value `"VNC"` and `resourceStatus` value `"NA"`. This allows the MirrorLink Client to e.g. provide an application launch short cut, to enable the user to directly launch a particular application, without having the user going through the Home Screen Application.

[024-009] – Chapter 2.5.5, Page 24, Line 4 – Use of Application Certificate Info

CLARIFICATION (2015-03-18) – The specification is speaking about the use of the GetCertifiedApplictionsList and GetApplicationCertificateInfo actions, but clarification is needed, that use latter action only, needs further actions of the MirrorLink server.

Therefore the text has to be extend by:

> The `GetApplicationCertificateInfo` action provides certification data for one application. Certification data is provided, if the application certificate is successfully validated by the MirrorLink Client and has any certifying entity, independent of whether the certifying entities include a CCC- and/or any Member-certifying entity section.
>
> The MirrorLink Clients will use the received information to decide, whether the application is certified under its conditions (e.g. locale, member certification, drive/base certification). The MirrorLink Client MAY use the `GetCertifiedApplicationsList` action and the `AppCertFilter` settings to pre-filter applications, as described in section 3.4.

[024-010] – Chapter 3.3, Page 32, Line 1-4 – Return optional element in AppList

CLARIFICATION (2015-03-18) – The specification is speaking about which A_ARG_TYPE_AppList elements need to be included in the response to an UPnP GetApplicationList action. Optional elements are not required to be returned unless being request. But the App Certification Handling specification at the same time requires the MirrorLink Server to include all elements from the certificate into the listing.

Therefore the text has to be extend by:

> Elements and attributes REQUIRED by the `A_ARG_TYPE_AppList` schema to be present in the `AppListing` output argument are always returned. Optional elements and attributes MUST be returned, regardless of their values, if the `AppListingFilter` explicitly specifies the OPTIONAL element or attribute to be returned (as shown in Example 1). The MirrorLink Server MUST include all optional elements into the response, which are provided from the MirrorLink Application Certificate or otherwise required from the Application Certificate Handling Specification (e.g. the `trustLevel` values).

[024-011] – Chapter 2.2.11, Page 16, Table 2-8 – Definition of restricted/nonRestricted

CLARIFICATION (2015-03-18) – The specification shows the restricted and nonRestricted elements of the A_ARG_TYPE_AppCertificateInfo structure as being defined outside the specification (using the +). But the locales are defined within the specification.

Therefore the text has to be changed to (the + symbol is removed):

Element	Description	Parent	Availability
restricted~~+~~	Comma separated list of locales, where restricted use is certified (A_ARG_TYPE_String)	entity	Required
nonRe-stricted~~+~~	Comma separated list of locales, where non-restricted use is certified (A_ARG_TYPE_String)	entity	Required

[024-012] – Chapter 3.4, Page 32, Line 11-15 – Usage of AppCertFilter

CLARIFICATION (2015-03-18) – The specification is not very clear on how to handle AppCert-Filter settings with multiple elements, in particular when the MirrorLink Client is looking for CCC or member-certified application, certified for a specific locale.

Therefore the following text is extended:

> *AppCertFilter* is composed of a comma-separated list of *A_ARG_TYPE_AppCertificateInfo* schema elements, attributes and their values. For clarification, the commas in the filter expression are equivalent to an AND operation. ~~For example, "name="CCC",restricted="*"" is interpreted as the condition "name="CCC"" AND "restricted="*"".~~ In case all comma-separated elements within the filter expression belong to the same parent, then all given conditions MUST apply for the same parent instance, in case multiple instances exist. The MirrorLink Server MUST apply the `AppCertFilter` only to the `entity` elements.
>
> Valid examples are given below:
>
> - An `AppCertFilter` set to "`name="CCC",restricted="EU"`" returns all applications, which are CCC drive-certified for EU.
> - An `AppCertFilter` set to "`name="CCC",restricted="USA"`" returns all applications, which are CCC drive-certified for USA.
> - An `AppCertFilter` set to "`name="CCC",restricted="JPN"`" returns all applications, which are CCC drive-certified for Japan.
> - An `AppCertFilter` set to "`name="OEM-A",targetList@target="*2011*"`" returns all applications, which are OEM-A member-certified and which include "`2011`" within any of the `target` elements as well.
>
> Optional as well as mandatory schema elements can be included in the `AppCertFilter`.

[024-013] – Chapter 2.2.11, Page 17, Line 38 – List of locales extensible

CLARIFICATION (2015-03-18) – The list of locales is not fixed and may be extended in the future to include other locales.

Therefore the following text is extended:

> The list of locales MAY be extended in future versions of the specification. The localities MUST be used case-insensitive, when comparing them with other values.

[024-014] – Chapter 2.2.4, Page 9, Line 24 – Wildcard in A_ARG_TYPE_AppID

CLARIFICATION (2015-06-17) – The A_ARG_TYPE_AppID is defined as a 32-bit integer in hexadecimal format, including a 0x pre-fix. At the same time we allow the use of a wildcard operator "*" at least for the GetApplicationStatus action. The wildcard would violate the format.

Therefore we need to add a separate note:

> An `A_ARG_TYPE_AppID` value MAY be identical to the wildcard "*", but it MUST NOT be used, unless its usage is specifically stated in the definition of the respective UPnP actions and/or events.

CCC-TS-026: UPnP Client Profile Service

The overview of Errata changes are listed below. Those reference the 1.1.4 version of the UPnP Client Profile Service specification (CCC-TS-026).

Identifier	Description	Type	Status
[026-001]	Uniqueness of Client ID	Clarification	Approved, 18.03.2015
[026-002]	HW/SW revision in modelNumber	Clarification	Approved, 18.03.2015
[026-003]	Client Profile manufacturer entry mandatory	Clarification	Approved, 18.03.2015
[026-004]	Make modelNumber mandatory	Clarification	Approved, 18.03.2015

The individual errata changes are specified below in detail.

[026-001] – Chapter 2.2.2, Table 2.2, Page 7 – Uniqueness of Client ID

CLARIFICATION (2015-03-18) – The UPnP Client Profile XML structure contains a Client ID value, which allows the MirrorLink Server to identify the connected MirrorLink Client, also beyond a MirrorLink session. Therefore the full description of the `clientID` element is:

> ID of the MirrorLink Client
>
> Unique identifier of the MirrorLink Client device instance; this identifier MUST be unique for all device instances from the device manufacturer given in the `manufacturer` element[1]. It MUST survive the MirrorLink Client's shut down and reboot. If MAY be reset during a factory reset.
>
> Note: The `ClientID` of some older MirrorLink Client devices MAY NOT be unique for all device instances.
>
> (A_ARG_TYPE_String)

[026-002] – Chapter 2.2.2, Table 2.2, Page 7 – HW/SW revision in modelNumber

CLARIFICATION (2015-03-18) – The Certification WG has recommend TWG to add elements into the MirrorLink 1.1 specification to better allow connected MirrorLink devices to identify the device's SW/HW revision. This would allow to provide a revision dependent handling of potential field failures, in particular of devices already out in the market. Additionally, it will also help CWG in its field failure process. Such a mechanisms will also help in case future problems occur on the ACMS interface.

The UPnP Client Profile XML is supporting a `modelNumber` element, which seem to be appropriate for such content. Therefore the full description of the `modelNumber` element is:

> Model Number
>
> Unique number identifying a family of devices, which expose identical MirrorLink related behavior, from the device manufacturer given in the `manufacturer` element. The model number format is vendor specific. It MUST be smaller than 32 bytes. The `modelNumber` values are recorded by the CCC Certification Body.
>
> Note: Some older MirrorLink Client devices MAY NOT provide a model number.

[1] Note: This is effectively a serial number, maintained by the Client vendor.

(A_ARG_TYPE_String)

Therefore the availability of the `modelNumber` element is:

~~Optional~~

Required

[026-003] – Chapter 2.2.2, Table 2.2, Page 7 – Client Profile manufacturer entry mandatory

CLARIFICATION (2015-03-18) – The UPnP Client Profile XML structure contains a manufacturer entry, which allows the MirrorLink Server to identify the connected MirrorLink Client vendor. The mandatory client ID is unique for a given Client Vendor, therefore the manufacturer has to be provided as well to allow the Server to make use of this information.

Therefore the updated obligation is:

Element	Description	Parent	Availability
manufacturer	Manufacturer Name (A_ARG_TYPE_String)	clientProfile	~~Optional~~ Mandatory

[030-009] – Chapter 4, Page 19, Line 15 – Make modelNumber mandatory

CLARIFICATION (2015-03-18) – The UPnP Client Profile XML structure contains a manufacturer entry, which allows the MirrorLink Server to identify the connected MirrorLink Client vendor. The mandatory client ID is unique for a given Client Vendor, therefore the manufacturer has to be provided as well to allow the Server to make use of this information.

Therefore the UPnP Client Profile XSD has to be updated:

```
<xs:element name="modelNumber" type="xs:string" minOccurs="1~~0~~"/>
```

CCC-TS-028: UPnP Notification Server Service

The overview of Errata changes are listed below. Those reference the 1.1.4 version of the UPnP Notification Server Service specification (CCC-TS-028).

Identifier	Description	Type	Status
[028-001]	No action list in A_ARG_TYPE_Notification	Clarification	Approved, 17.06.2014
[028-002]	Handling of missing action list	Clarification	Approved, 17.06.2014

The individual errata changes are specified below in detail.

[028-001] – Chapter 2.3.4, Table 2.2, Page 11 – No action list in A_ARG_TYPE_Notification

CLARIFICATION (2014-06-17) – The action list element in the A_ARG_TYPE_Notification structure is optional. Whereas the use of the action list has been described, text is missing, how the MirrorLink Client will handle any missing action list. Therefore the full description of the `actionList` element is:

> A list of actions for a notification.
>
> The list is provided by an application initiating the notification so a user can directly select one of those actions for the notification.
>
> For example, the user can "Reply" to the new text message or "Ignore" it. The list includes "Reply" and "Ignore" actions as its elements.
>
> A MirrorLink Client SHOULD launch the application with provided `appID`, without showing any notification, if the `actionList` element is missing.

[028-002] – Chapter 3.2.2, Page 22, Line 21 – Handling of missing action list

CLARIFICATION (2014-06-17) – The action list element in the A_ARG_TYPE_Notification structure is optional. The steps in Figure 4 describe the use of the action list, but text is missing, how the MirrorLink Client will handle any missing action list. Therefore a sentence is added after Figure 4 to describe the behavior:

> The MirrorLink Client will skip steps 3 - 5, if the Notification does not contain an action list and will immediately launch the application with the given <appID> value.

CCC-TS-030: UPnP Server Device

The overview of Errata changes are listed below. Those reference the 1.1.4 version of the UPnP Server Device specification (CCC-TS-030).

Identifier	Description	Type	Status
[030-001]	Description of X_Signature element	Bug Fix	Approved, 22.01.2014
[030-002]	Reference URI	Bug Fix	Approved, 22.01.2014
[030-003]	Canonicalization	Clarification	Approved, 22.01.2014
[030-004]	Xml:id attribute in Example	Bug Fix	Approved, 18.03.2014
[030-005]	Reference URI in Example	Bug Fix	Approved, 18.03.2014
[030-006]	Transform in Example	Bug Fix	Approved, 18.03.2014
[030-007]	KeyInfo in Example	Clarification	Approved, 18.03.2014
[030-008]	HW/SW revision in modelNumber	Clarification	Approved, 18.03.2015
[030-009]	Make modelNumber mandatory	Clarification	Approved, 18.03.2015
[030-010]	Mandating Device XML order	Clarification	Approved, 17.06.2015

The individual errata changes are specified below in detail.

[030-001] – Chapter 2, Page 10 – Description of X_Signature element in Table 3

BUG FIX (2014-01-22) – The Signature is calculated over the root document and not over the device element. Correct text is:

> XML signature over entire contents of the ~~device~~ `root` element. […]
>
> The Reference element of the XML signature MUST be empty ~~point to device element~~.

[030-002] – Chapter 2.3.1, Page 10, Line 16-18 – Reference URI

BUG FIX (2014-01-22) – The Signature is calculated over the root document and not over the device element. Correct text is:

> The Reference URI MUST be empty ~~NOT be outside document. It MUST refer to the <device> element, which is the parent of the <ml11:X_Signature> element, for the UPnP Server device description. When the URI attribute is omitted, empty or of an unknown format for the MirrorLink Client to recognize, the MirrorLink Client MUST refer to the element listed above~~.

[030-003] – Chapter 2.3.1, Page 11, Line 2 – Canonicalization

CLARIFICATION (2014-01-22) – Clarification that Canonical version 1.0 includes xml-c14n and xml-exc-c14n. Correct text is:

> The MirrorLink Server MUST use Canonical method XML version 1.0 (`xml-c14n` or `xml-exc-c14n`) or 1.1 (xml-c14n11);

[030-004] – Chapter 3, Page 12, Line 28 – xml:id attribute in Example

BUG FIX (2014-03-18) – No use of the xml:id attribute due to W3C recommendation in conjunction with C14N canonicalization. Correct text is:

> `<device ~~xml:id="mlServerDevice"~~>`

[030-005] – Chapter 3, Page 14, Line 39 – Reference URI in Example

BUG FIX (2014-03-18) – No use of the xml:id attribute due to W3C recommendation in conjunction with C14N canonicalization. Correct text is:

```
<Reference URI="#mlServerDevice">
```

[030-006] – Chapter 3, Page 14, Line 42 – Transform in Example

BUG FIX (2014-03-18) – Transformation missing for enveloped-signature. Add the following line(s) after line 42

```
<Transform Algorithm=
"http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
```

[030-007] – Chapter 3, Page 14, Line 53 to Page 15, Line 4 – KeyInfo element in Example

CLARIFICATION (2014-03-18) – MirrorLink Client SHALL NOT use the Key Info entry in the Signature for validation purpose. Delete the following lines

```
<KeyInfo>
  <KeyValue>
    <DSAKeyValue>
      <P>...</P>
      <Q>...</Q>
      <G>...</G>
      <Y>...</Y>
    </DSAKeyValue>
  </KeyValue>
</KeyInfo>
```

[030-008] – Chapter 2.3, Page 10, Line 2 – HW/SW revision in modelNumber

CLARIFICATION (2015-03-18) – The Certification WG has recommend TWG to add elements into the MirrorLink 1.1 specification to better allow connected MirrorLink devices to identify the device's SW/HW revision. This would allow to provide a revision dependent handling of potential field failures, in particular of devices already out in the market. Additionally, it will also help CWG in its field failure process. Such a mechanisms will also help in case future problems occur on the ACMS interface.

The UPnP Server Device XML is supporting a `modelNumber` element, which seem to be appropriate for such content. Therefore the following description of the `modelNumber` element need to be added:

> The `modelNumber` element within the Device XML is a unique number identifying a family of devices, which expose identical MirrorLink related behavior, from the device manufacturer given in the `manufacturer` element. The model number format is vendor specific. It MUST be smaller than 32 bytes. The `modelNumber` values are recorded by the CCC Certification Body.
>
> Note: Some older MirrorLink Server devices MAY NOT provide a model number.

[030-009] – Appendix A, Page 19, Line 16 – Make modelNumber mandatory

CLARIFICATION (2015-03-18) – The Certification WG has recommend TWG to add elements into the MirrorLink 1.1 specification to better allow connected MirrorLink devices to identify the device's SW/HW revision. This would allow to provide a revision dependent handling of potential field failures, in particular of devices already out in the market. Additionally, it will also help CWG in its field failure process. Such a mechanisms will also help in case future problems occur on the ACMS interface.

The UPnP Server Device XML is supporting a `modelNumber` element, which seem to be appropriate for such content. Therefore the following the `modelNumber` element need to be mandatory.

Therefore the Device XSD has to be updated:

```
<xsd:element name="modelNumber" minOccurs="1~~0~~">
```

[030-010] – Chapter 3, Page 12, Line 7-20 – Mandating Device XML order

CLARIFICATION (2015-06-17) – UPnP Device Architecture 1.1 requires the UPnP Server Device XML to maintain its order. Device Architecture 1.0 only provides a recommendation. ML Server's so far seem have kept the Device XML order. In order to prevent problems with ML Clients, we need to extend the order to both device architecture versions.

> The MirrorLink Server MUST provide a well-formed XML and a correct parent children relationship of xml elements and correct xml namespaces URI for each element. If the MirrorLink Server provides MirrorLink extension elements (X_...) then those elements ~~SHOULD~~ MUST be valid according their XSD provided in Appendix A. ~~If t~~The MirrorLink Server ~~declares conformance to UPnP DA 1.1 in the `specVersion` element then it~~ MUST provide XML Device Description that is valid according to XSD presented in Appendix A. ~~Otherwise the MirrorLink Server SHOULD provide XML Device Description that is valid according to XSD presented in Appendix A.~~
>
> The MirrorLink Client SHOULD be as permissive as possible when consuming XML.
>
> As a general recommendation for better interoperability, the MirrorLink Servers SHOULD ~~use the order of XML elements presented in the listing below, and~~ declare XML namespaces as default to allow interoperability with Clients without XML namespaces support. MirrorLink Servers ~~SHOULD~~ MUST add any element unspecified below after the specified ones and put it in a namespace that isn't specified below. MirrorLink Clients SHOULD be able to handle elements in wrong or any xml namespace if the element wasn't found in the correct one.

CCC-TS-032: CORE ARCHITECTURE

The overview of Errata changes are listed below. Those reference the 1.1.0a version of the Core Architecture specification (CCC-TS-032).

Identifier	Description	Type	Status
[032-001]	DAP optional for ML 1.0	Bug Fix	Approved, 10.11.2014

The individual errata changes are specified below in detail.

[032-001] – Chapter 4, Page 11 – DAP optional for ML 1.0 in Table 1

BUG FIX (2014-11-10) – The support for a DAP Server is optional for MirrroLink 1.0. It is only mandatory from ML 1.1 onwards. Correct table is:

Feature			Version	MirrorLink Server	MirrorLink Client
[…]			[…]	[…]	[…]
Security	DAP	Server Endpoint	1.0.1~~, 1.1~~	~~MUST~~SHOULD	N/A
			1.1	MUST	N/A
		Client Endpoint	1.0.1, 1.1	N/A	SHOULD

Applicable for MirrorLink 1.2 Specification: YES

CCC-TS-036: HANDLING OF APPLICATION CERTIFICATES

The overview of Errata changes are listed below. Those reference the 1.1.11 version of the Handling of Application Certificates specification (CCC-TS-036).

- [036-001] – [036-003] have been integrated into version 1.1.10

Identifier	Description	Type	Status
[036-004]	Platform and Runtime validation	Clarification	Approved, 17.06.2015
[036-005]	Query Timer Reset	Clarification	Approved, 17.06.2015

The individual errata changes are specified below in detail.

[036-004] – Chapter 4.1.2, Page 21, Line 10-14 – Platform and Runtime validation

CLARIFICATION (2015-06-17) – The specification states that a ML Server MUST retry to download a certificate, if the application is not certified for the platform and the runtime. This actually includes both, the platform/runtime and the platform/runtime version.

Therefore the following text is amended:

> The MirrorLink Server MUST retry to download a new application certificate between 50% and 100% of the query period after the last HTTP-Get attempt in case the following validation steps failed:
>
> - Application certificate is expired
> - Application is not certified for the MirrorLink Server's platform or the platform version is blacklisted.
> - Application is not certified for the MirrorLink Server's runtime or the runtime version is blacklisted.

[036-005] – Chapter 4.3.1, Page 27, Line 26-27 – Query Timer Reset

CLARIFICATION (2015-06-17) – The specification hints that the Query timer must be reset after receiving a successful OCSP response. But a more explicit statement should avoid doubts.

Therefore the following text is amended:

> The MirrorLink Server MUST NOT start the query period timer, prior the first connection establishment to a MirrorLink Client. The MirrorLink Server MUST reset the query period if it has received a valid OCSP response, whose *certStatus* is "good" and which has been successfully validated.

REFERENCES

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