ETSI TS 103 544-5 V1.3.1 (2019-10)

TECHNICAL SPECIFICATION

Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink®; Part 5: Common Data Bus (CDB)

CAUTION

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

ETSI

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

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

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

Important notice

The present document can be downloaded from:
http://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at www.etsi.org/deliver.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

Copyright Notification

ETSI

Contents

Intellectual Property Rights

Essential patents

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

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

Trademarks

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

Foreword

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

The present document is part 5 of a multi-part deliverable. Full details of the entire series can be found in part 1 [i.1].

Modal verbs terminology

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

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

1 Scope

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

The Common Data Bus (CDB) is a simple, low-bandwidth shared bus, which allows exchanging data between two CDB endpoints, residing in the MirrorLink Server and Client. The Common Data Bus is fully symmetrical, i.e. services can be provided on both endpoints independently from each other.

2 References

2.1 Normative references

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

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

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

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

[1] ETSI TS 103 544-9 (V1.3.1): "Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink®; Part 9: UPnP Application Server Service".

2.2 Informative references

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

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

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

[i.1] ETSI TS 103 544-1 (V1.3.1): "Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink®; Part 1: Connectivity".

3 Definition of terms, symbols and abbreviations

3.1 Terms

Void.

3.2 Symbols

Void.

3.3 Abbreviations

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

API	Application Programming Interface
CCC	Car Connectivity Consortium
CDB	Common Data Bus
DAP	Device Attestation Protocol
IP	Internet Protocol
TCP	Transmission Control Protocol
UPnP	Universal Plug-and-Play
URL	Universal Resource Locator
VNC	Virtual Networking Computing

4 Common Data Bus Architecture

4.1 General Architecture

The Common Data Bus (CDB) is a simple, low-bandwidth shared bus, which allows exchanging data between two CDB endpoints, residing in the MirrorLink Server and Client. The Common Data Bus is fully symmetrical, i.e. services can be provided on both endpoints independently from each other.

A CDB endpoint can host a CDB Sink endpoint and a CDB Source endpoint. CDB Sink endpoints are subscribing to data services provided from CDB Source endpoints. A CDB source endpoint can provide multiple data services from data sources. A CDB sink endpoint can deliver data from multiple data sources to multiple data sinks. The endpoints are responsible for marshalling and de-marshalling of all the data from multiple applications passing through the common data bus. A CDB data sink subscribes to a service, provided from a data source service. For simplicity, the following figure shows two CDB services, which one data source and data sink each.

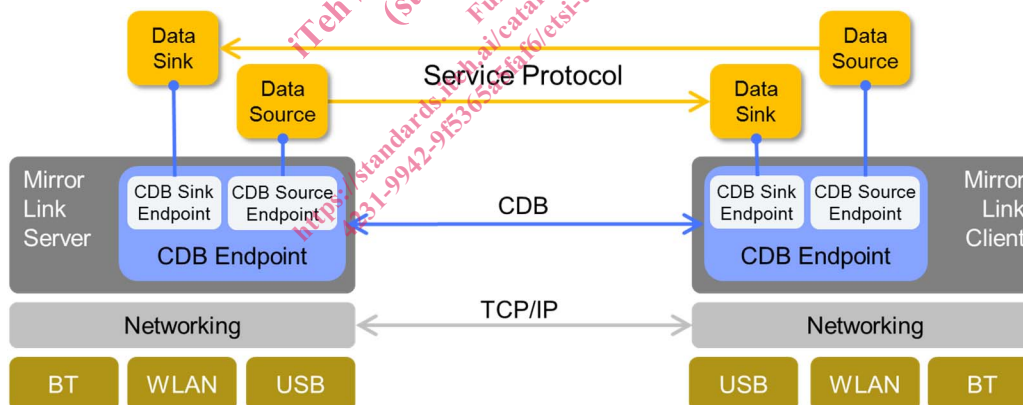**Figure 1: Common Data Bus Architecture with TCP binding**

4.2 Bindings

4.2.1 TCP Binding

4.2.1.1 Launching the Common Data Bus

The MirrorLink Server, providing Common Data Bus functionality, shall include the Common Data Bus into its application listing as specified in [1]. The MirrorLink Client shall identify the Common Data Bus from the application listing as specified in [1].

The Common Data Bus shall be started from the MirrorLink Client, using the UPnP *TmApplicationServer*:1 service *LaunchApplication* action [1]. The MirrorLink Client's CDB endpoint shall open a TCP connection to the Server's CDB endpoint, using the returned URL from the *LaunchApplication* action.

The MirrorLink Client should launch the MirrorLink Server's CDB endpoint not later than 10s after receiving the *first A_ARG_TYPE_AppList* response from the MirrorLink Server.

The MirrorLink Client shall have launched the MirrorLink Server's CDB prior starting the first VNC based remote application.

4.2.1.2   Intentionally Terminating the Common Data Bus

The MirrorLink Client and Server can terminate the Server's Common Data Bus endpoint anytime, using the CDB *ByeBye* message and the UPnP *TmApplicationServer:1* services, as defined in [1].

The CDB endpoint in the MirrorLink Client shall use the following sequence to terminate the CDB operation:

1) Client CDB endpoint shall send a CDB *ByeBye* message. The CDB Client shall not send any further CDB messages, after sending the CDB *ByeBye* message. The CDB Client endpoint should ignore all incoming CDB messages, after sending a CDB *ByeBye* message.

2) Server CDB endpoint shall respond with a CDB *ByeBye* message.

3) Client CDB endpoint shall disconnect the TCP connection. The CDB Client should disconnect the TCP connection, if it does not receive the CDB *ByeBye* message back within 5 s.

4) CDB Server should disconnect the TCP connection on detection of the Client TCP disconnect or 5 s after sending the CDB *ByeBye* message, whatever comes first.

Client CDB endpoint should send a UPnP *TmApplicationServer*:1 service *TerminateApplication* action for the server CDB endpoint.

The CDB endpoint in the MirrorLink Server shall use the following sequence to terminate the CDB operation:

1) Server CDB endpoint shall send a CDB *ByeBye* message. The Server CDB endpoint shall not send any further CDB messages after sending the CDB *ByeBye* message. The CDB endpoint should ignore all incoming CDB messages, after sending a CDB *ByeBye* message.

2) Client CDB endpoint shall disconnect the TCP connection.

3) Server CDB endpoint shall signal the CDB endpoint's termination to the Client, if it has subscribed to the *TmApplicationServer:1 AppStatusUpdate* event.

4) Server CDB endpoint should disconnect the TCP connection on detection of the Client TCP disconnect or 5s after sending the CDB *ByeBye* message, whatever comes first.

If the CDB is terminated prior to the establishment of the TCP connection, steps 1, 2 and 4 shall be omitted.

4.2.1.3   Unintentionally Terminating the CDB Session

Unintentional termination of the CDB session may happen any time in case of error conditions. In this case the respective CDB Server or Client endpoint will disconnect the TCP connection. The respective counterpart should disconnect as well.

If the MirrorLink Client decides to re-establish the CDB session, it shall follow the steps given in clause 4.2.1.1.

To avoid the CDB Server or Client endpoint being in a TCP TIME-WAIT time-out loop, as a result of an unintentional active disconnect, the TCP socket should be established using the SO_REUSEADDR option (or similar platform specific variants), allowing the operating system to reuse a port address, even it is currently in the TIME-WAIT state or the CDB Server endpoint should use a different, unaffected port.

4.2.2 Other Bindings

Besides TCP/IP, it will be also possible to run MirrorLink Common Data Bus on top of other protocol like Bluetooth RFCOMM, but how to discover and establish connection for such configuration is outside the scope of the present document.

4.3 Testing Considerations

If the MirrorLink Client is in a dedicated testing state (as part of the MirrorLink Certification), it shall launch a new CDB session (either initiated automatically or manually from the user), whenever the CDB Server endpoint has intentionally terminated the CDB session.

If the MirrorLink Client is in a dedicated testing state (as part of the MirrorLink Certification), it shall launch a new CDB session (either initiated automatically or manually from the user), whenever the CDB Server endpoint has unintentionally terminated the CDB session.

5 Message Types and Format

5.1 Message Overview

The Common Data Bus (CDB) defines the following messages, which are specified in more detail in the following paragraphs:

- *ServicesRequest*: Requests the list of supported services

- *ServicesSupported*: Provide a list of supported data services

- *StartService*: Request to start a specific service

- *StopService*: Request to stop a specific service

- *ServicePayload*: Deliver service specific payload

- *ServiceResponse*: Response to *StartService*, *StopService*, *ServicePayload*

- *ByeBye*: Terminates the Common Data Bus

- *Ping*: Message to check the connection

- *PingResponse*: Responds to a Ping message

All U16 values are encoded in big endian.

5.2 ServicesRequest

The *ServicesRequest* message is used from the CDB Sink endpoint to request the list of supported services from the CDB Source endpoint.

Table 1: *ServicesRequest* Message

# bytes	Type	Value	Description
2	U16	0xB101	Message-type
2	U16	2	Payload length
1	U8	1	CDB sink endpoint major version
1	U8	1	CDB sink endpoint minor version

After the Common Data Bus connection is initiated between the 2 CDB endpoints, each CDB sink endpoint can send a *ServicesRequest* message to indicate that it is interested in getting the list of services available at the CDB source endpoint and is also interested in getting updates whenever the list changes.

If a CDB endpoint supports CDB sink functionality, the CDB Sink endpoint shall send a *ServicesRequest* message within 5 s after the CDB connection has been established.

The CDB Source endpoint shall respond with a *ServicesSupported* message to the *ServicesRequest* message within 5 s. The CDB Sink endpoint should not send a *ServicesRequest* message, if it does not support service subscription.

In case a response is not received within 5 s, the CDB Sink endpoint shall assume that the CDB Source is not providing any service. No further action is required.

Implementation Note:

> The present document defines a CDB sink endpoint version of 1.1. Note that the CDB sink endpoint version is not necessarily identical with the MirrorLink version. The following CDB versions are supported within the different MirrorLink versions:

- MirrorLink 1.1: CDB sink endpoint version 1.1.

- MirrorLink 1.2: CDB sink endpoint version 1.1.

- MirrorLink 1.3: CDB sink endpoint version 1.1.

5.3 ServicesSupported

The *ServicesSupported* message shall be used from the CDB Source endpoint to notify the CDB Sink endpoint about the data services it can provide.

Table 2: *ServicesSupported* Message

# bytes	Type	Value	Description
2	U16	0xB102	Message-type
2	U16	4+M	Payload length
1	U8	1	CDB source endpoint major version
1	U8	1	CDB source endpoint minor version
2	U16	N	Total number of Services
M	Array of U8		Array of service descriptions, as defined in Table 3.

The CDB Source endpoint version shall be equal or smaller than the CDB Sink endpoint's version. Otherwise the CDB Sink endpoint shall send a *ServiceResponse* message with the Error code 0x0209 in response to the received *ServicesSupported* message.

Implementation Note:

> The present document defines a CDB source endpoint version of 1.1. Note that the CDB source endpoint version is not necessarily identical with the MirrorLink version. The following CDB versions are supported within the different MirrorLink versions.

- MirrorLink 1.1: CDB source endpoint version 1.1.

- MirrorLink 1.2: CDB source endpoint version 1.1.

- MirrorLink 1.3: CDB source endpoint version 1.1.

Each service is described as given in Table 3.