# Car Connectivity Consortium MirrorLink®

## **Common Data Bus**

Version 1.1.3 (CCC-TS-016)



Copyright © 2011-2013 Car Connectivity Consortium LLC

All rights reserved

Confidential

# **VERSION HISTORY**

Version	Date	Comment
1.1	31 March 2012	Approved Version
1.1.1	24 September 2012	Approved Errata Version
1.1.2	05 March 2013	Approved Errata Version
1.1.3	05 November 2013	Approved Errata Version

# CONTRIBUTORS

2

4	Bose, Raja	Nokia Corporation
5	Brakensiek, Jörg (Editor)	Nokia Corporation
6	Park, Keun-Young	Nokia Corporation

### LEGAL NOTICE

- 2 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
- 4 by these license terms and the CCC LLC Limited Liability Company Agreement (the "Agreement").
- 5 Use of the Specification by anyone who is not a member of CCC LLC (each such person or party, a
- 6 "Member") is prohibited. The legal rights and obligations of each Member are governed by the Agreement
- 7 and their applicable Membership Agreement, including without limitation those contained in Article 10 of
- 8 the LLC Agreement.
- 9 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 ("Im-
- plementing Products") under the terms of the Agreement (the "Purpose"). Members are not permitted to
- 12 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
- 14 subcontractors have a need to know for carrying out the Purpose and provided that such Affiliates and
- 15 subcontractors accept confidentiality obligations similar to those contained in the Agreement. Each Mem-
- ber shall be responsible for the observance and proper performance by such of its Affiliates and subcon-
- 17 tractors of the terms and conditions of this Legal Notice and the Agreement. No other license, express
- 18 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
- 20 Membership Agreement is prohibited and any such prohibited use may result in termination of the appli-
- 21 cable Membership Agreement and other liability permitted by the applicable Agreement or by applicable
- 22 law to CCC LLC or any of its members for patent, copyright and/or trademark infringement.
- 23 THE SPECIFICATION IS PROVIDED "AS IS" WITH NO WARRANTIES, EXPRESS OR IMPLIED,
- 24 INCLUDING WITHOUT LIMITATION ANY WARRANTY OF MERCHANTABILITY, FITNESS FOR A
- 25 PARTICULAR PURPOSE, NONINFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL
- 26 PROPERTY RIGHTS, AND COMPLIANCE WITH APPLICABLE LAWS.
- 27 Each Member hereby acknowledges that its Implementing Products may be subject to various regulatory
- 28 controls under the laws and regulations of various jurisdictions worldwide. Such laws and regulatory
- 29 controls may govern, among other things, the combination, operation, use, implementation and distribu-
- 30 tion of Implementing Products. Examples of such laws and regulatory controls include, but are not limited
- 31 to, road safety regulations, telecommunications regulations, technology transfer controls and health and
- 32 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
- 34 licenses for their Implementing Products related to such regulations within the applicable jurisdictions.
- 35 Each Member acknowledges that nothing in the Specification provides any information or assistance in
- 36 connection with securing such compliance, authorizations or licenses.
- 37 NOTHING IN THE SPECIFICATION CREATES ANY WARRANTIES, EITHER EXPRESS OR IMPLIED,
- 38 REGARDING SUCH LAWS OR REGULATIONS. ALL LIABILITY, INCLUDING LIABILITY FOR
- 39 INFRINGEMENT OF ANY INTELLECTUAL PROPERTYRIGHTS OR FOR NONCOMPLIANCE WITH
- 40 LAWS, RELATING TO USE OF THE SPECIFICATION IS EXPRESSLY DISCLAIMED. BY USE OF
- 41 THE SPECIFICATION, EACH MEMBER EXPRESSLY WAIVES ANY CLAIM AGAINST CCC LLC AND
- 42 ITS MEMBERS RELATED TO USE OF THE SPECIFICATION.
- 43 CCC LLC reserve the right to adopt any changes or alterations to the Specification as it deems necessary
- 44 or appropriate.
- 45 Copyright © 2011-2013. CCC LLC.

# **TABLE OF CONTENTS**

2	VERSI	ON HISTORY	2
3	CONTI	RIBUTORS	2
4	LEGAI	L NOTICE	3
5	TABLE	OF CONTENTS	4
6	TERMS	S AND ABBREVIATIONS	5
7	1 IN	TRODUCTION	6
8	2 C(	OMMON DATA BUS ARCHITECTURE	7
9	2.1	GENERAL ARCHITECTURE	7
10	2.2	BINDINGS	7
11	2.2	2.1 TCP Binding	
12	2.2		8
13	2.3	TESTING CONSIDERATIONS	8
14	3 M	ESSAGE TYPES AND FORMAT	
15	3.1	ServicesRequest	10
16	3.2	SERVICESSUPPORTED	
17	3.3	STARTSERVICE	
18	3.4	STOPSERVICE	
19	3.5	ServicePayload	
20	3.6	SERVICERESPONSE	
21	3.7	BYEBYE	
22	3.8	Ping	
23	3.9	PINGRESPONSE	
24	4 M	ESSAGE FLOW	16
25	5 RI	EFERENCES	18
-			

# TERMS AND ABBREVIATIONS

2	BT	Bluetooth
3	CDB	Common Data Bus
4	IP	Internet Protocol
5	MMU	Memory Management Unit
6	OOB	Out-of-Band
7	TCP	Transmission Control Protocol
8	UDP	User Datagram Protocol
9	UI	User Interface
10	UPnP	Universal Plug-and-Play
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		
25		
26		
27		
28		
29		trademark of Car Connectivity Consortium LLC.
30		egistered trademark of Bluetooth SIG Inc.
31	UPnP is a regist	ered trademark of UPnP Forum.

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

#### 1 Introduction

- 2 This document is part of the MirrorLink specification, which specifies an interface for enabling remote user
- 3 interaction of a mobile device via another device. This specification is written having a car head-unit to in-
- 4 teract with the mobile device in mind, but it will similarly apply for other devices, which do provide a colored
- 5 display, audio input/output and user input mechanisms.
- 6 This specification describes the Common Data Bus, a data exchange mechanism which can be utilized by
- 7 applications running on the MirrorLink platform. The Common Data Bus is a simple extensible transport
- 8 mechanism which:

1

9

10

12

13 14

15

16

30

31

32

33

34

35

36 37

38

39

40

41

42

43 44

- Provides a common bus between two end-points for bi-directional exchange of data over the same transport medium.
- Provides a light-weight way to host services on a client.
  - Useful for applications which do not have heavy bandwidth requirements.
  - Avoids use of multiple network socket connections for each independent client-server application thereby reducing processing and memory overhead and improving performance.
    - Has low implementation overhead.
    - Adheres to a simple yet functionally complete protocol.
- Is application agnostic.
- Makes no assumption about APIs utilized by either end-point to communicate locally with their own
   hosted applications.
- In this document we cover the details of the Common Data Bus system architecture, message types, messaging protocol, examples of services which can utilize it and error recovery.
- The specification lists a series of requirements, either explicitly or within the text, which are mandatory ele-
- 23 ments for a compliant solutions. Recommendations are given, to ensure optimal usage and to provide suitable
- 24 performance. All recommendations are optional.
- 25 The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD",
- 26 "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are following the no-
- tation as described in RFC 2119 [2].
- 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 COMMON DATA BUS ARCHITECTURE

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

#### 2.1 General Architecture

- A CDB endpoint can host a CDB Sink endpoint and a CDB Source endpoint. CDB Sink endpoints are sub
  - scribing to data services provided from CDB Source endpoints. A CDB source endpoint can provide multiple
- 8 data services from data sources, A CDB sink endpoint can deliver data from multiple data sources to multiple
- 9 data sinks. The endpoints are responsible for marshalling and de-marshalling of all the data from multiple
- 10 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
- 12 and data sink each.

1

5

7

13

14

15

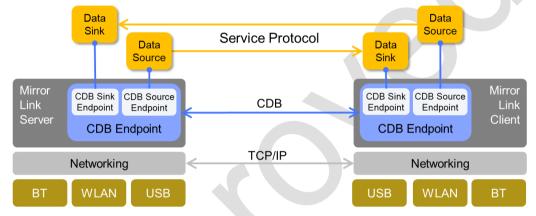


Figure 1: Common Data Bus Architecture with TCP binding

# 2.2 Bindings

Within this specification, the Common Data Bus has been defined with a TCP binding.

#### 17 2.2.1 TCP Binding

- 18 2.2.1.1 Launching the Common Data Bus
- 19 The MirrorLink server, providing Common Data Bus functionality, MUST include the Common Data Bus
- 20 into its application listing as specified in [1]. The MirrorLink Client MUST identified the Common Data Bus
- from the application listing as specified in [1].
- 22 The Common Data Bus MUST be started from the MirrorLink client, using the UPnP TmApplicationServer:1
- 23 service LaunchApplication action [1]. The MirrorLink client's CDB endpoint MUST open a TCP connection
- to the server's CDB endpoint, using the returned URL from the LaunchApplication action.
- 25 The MirrorLink client SHOULD launch the MirrorLink server's CDB endpoint not later than 10s after re-
- ceiving the first A\_ARG\_TYPE\_AppList response from the MirrorLink server.
- 27 The MirrorLink client MUST have launched the MirrorLink server's CDB prior starting the first VNC based
- 28 remote application.
- 29 2.2.1.2 Intentionally Terminating the Common Data Bus
- 30 The MirrorLink client and server can terminate the server's Common Data Bus endpoint anytime, using the
- 31 CDB ByeBye message and the UPnP TmApplicationServer:1 services, as defined in [1].

- 1 The CDB endpoint in the MirrorLink client MUST use the following sequence to terminate the CDB operation:
  - 1. Client CDB endpoint MUST send a CDB ByeBye message. The CDB client MUST 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 MUST respond with a CDB ByeBye message.
  - 3. Client CDB endpoint MUST disconnect the TCP connection. The CDB client SHOULD disconnect the TCP connection, if it does not receive the CDB ByeBye message back within 5s.
  - 4. CDB server SHOULD disconnect the TCP connection on detection of the client TCP disconnect or 5s after sending the CDB ByeBye message, whatever comes first.
- Client CDB endpoint SHOULD send a UPnP TmApplicationServer:1 service TerminateApplication action
- 12 for the server CDB endpoint.

4 5

6

7

8

10

- The CDB endpoint in the MirrorLink server MUST use the following sequence to terminate the CDB operation:
- Server CDB endpoint MUST send a CDB ByeBye message. The Server CDB endpoint MUST
   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.
- 18 2. Client CDB endpoint MUST disconnect the TCP connection
- Server CDB endpoint MUST 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.
- Note: If the CDB is terminated prior to the establishment of the TCP connection, steps 1, 2 and 4 MUST be omitted.
- 25 2.2.1.3 Unintentionally Terminating the CDB Session
- 26 Unintentional termination of the CDB session MAY happen any time in case of error conditions. In this case
- 27 the respective CDB server or client endpoint will disconnect the TCP connection. The respective counterpart
- 28 SHOULD disconnect as well.
- 29 If the MirrorLink Client decides to re-establish the CDB session, it MUST follow the steps given in Chapter
- 30 2.2.1.1.

39

- To avoid the CDB server or client endpoint being in a TCP TIME-WAIT time-out loop, as a result of an
- 32 unintentional active disconnect, the TCP socket SHOULD be established using the SO\_REUSEADDR option
- 33 (or similar platform specific variants), allowing the operating system to reuse a port address, even it is cur-
- rently in the TIME-WAIT state or the CDB server endpoint SHOULD use a different, unaffected port number.
- 35 2.2.2 Other Bindings
- 36 Besides TCP/IP, it will be also possible to run MirrorLink Common Data Bus on top of other protocol like
- 37 Bluetooth RFCOMM, but how to discover and establish connection for such configuration is outside the
- 38 scope of this specification.

## 2.3 Testing Considerations

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

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



#### 3 Message Types and Format

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

3 following paragraphs.

1

4

5

6

7

8

9

10

11

12

14

17

28

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

13 All U16 values are encoded in big endian.

#### 3.1 ServicesRequest

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

# bytes	Туре	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

Table 1: ServicesRequest Message

- After the Common Data Bus connection is initiated between the 2 CDB endpoints, each CDB sink endpoint
- 19 can send a ServicesRequest message to indicate that it is interested in getting the list of services avail-
- able at the CDB source endpoint and is also interested in getting updates whenever the list changes.
- 21 If a CDB endpoint supports CDB sink functionality, the CDB Sink endpoint MUST send a ServicesRe-
- 22 quest message within 5 s after the CDB connection has been established.
- 23 The CDB Source endpoint MUST respond with a ServicesSupported message to the ServicesRe-
- 24 quest message within 5s. The CDB Sink endpoint SHOULD NOT send a ServicesRequest message,
- 25 if it does not support service subscription.
- In case a response is not received within 5s, the CDB Sink endpoint MUST assume that the CDB Source is
- 27 not providing any service. No further action is required.

# 3.2 ServicesSupported

29 The ServicesSupported message MUST be used from the CDB Source endpoint to notify the CDB

30 Sink endpoint about the data services it can provide.

# bytes	Туре	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

# bytes	Туре	Value	Description
2	U16	N	Total number of Services
M	Array of U8		Array of service descriptions, as defined in Table 3.

Table 2: ServicesSupported Message

- 2 The CDB Source endpoint version MUST be equal or smaller than the CDB Sink endpoint's version. Other-
- 3 wise the CDB Sink endpoint MUST send a ServiceResponse message with the Error code 0x0209 in
- 4 response to the received ServicesSupported message.
- 5 Each service is described as given in the following table.

# bytes	Туре	Value	Description
2	U16		Unique Service ID
1	U8		Major version of service
1	U8		Minor version of service
		Bit	Service Configuration ('1' enabled, '0' disabled)
1	U8	[0]	Service Encryption Data payload is encrypted with the data provider's session key, exchanged as within the DAP of the server's CDB endpoint.
		[1]	Service Resource Constraints Only one service can be started, in case multiple entries with same service name, but different version, exist in the services supported list.
	U8	Bit	Service Access Control ('1' enabled, '0' disabled)
		[0]	Unlimited Data can be provided to any application.
		[1]	CCC-Certified Data can be provided to any application, certified from the Car Connectivity Consortium.
1		[2]	Source-Certified Data can be provided to any application, certified from the manufacturer of the source device.
		[3]	Service-Certified Data can be provided to any application, certified from the manufacturer of the source device to access the data service.
1	U8	N	Indicates the length of service name string.
N	Array of	U8	Service name string

Table 3: Service Description

Each service MUST have a service ID, which is unique during a CDB session. Service IDs for services provided from the MirrorLink server MUST have the range of 0x4001 to 0x7FFF, and service IDs for services

- 9 from the MirrorLink client MUST have the range of 0x0001 to 0x3FFF. The most significant bit is reserved
- 10 for future use and MUST be set to 0. The CDB Source endpoint MUST NOT change the Service ID number
- during a CDB session.

- 12 Setting more than one Service Access Control bit, will aggregate (i.e. OR) the access control conditions. The
- source grants no access to a data service, if all Service Access Control bits in the above table are set to zero
- 14 (0). The relationship between the Service Access Control layers is shown in Figure 2.
- 15 The manufacturer of the source device MUST be identified from the <manufacturer> entry in the UPnP
- 16 TmClientProfile:1 SetClientProfile action. If the client does not use the service, or the <manufacturer> entry

is left empty, the Source- and Service-Certified Service Access Control bits are ignored (i.e. assumed to be zero (0)).

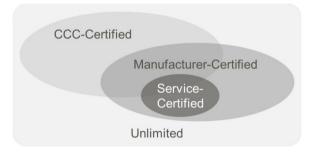


Figure 2: Service Access Control Layers

- 5 Service names MUST follow the naming convention domainName.serviceName, where domainName
- 6 is following the Java namespace convention (e.g. com.daimler). MirrorLink specified services MUST use
- 7 "com.mirrorlink" as the domain name.

3

4

17 18

19

20

21

- 8 A CDB Source endpoint MUST send a ServicesSupported message for every ServicesRequest
- 9 message that it receives. However, after the receipt of the first ServicesRequest message, a CDB Source
- 10 endpoint MUST send a Services Supported message, whenever the list of available services changes,
- even if the other endpoint has not sent any subsequent ServicesRequest messages.
- 12 The CDB source endpoint MUST list only one service with identical service name string and major version.
- Any CDB services MUST be backward compatible with regard to its minor versions.
- 14 A Services Supported message MUST contain the complete list of services currently available. In case
- 15 a CDB Source endpoint does not have any services available, it MUST respond with a ServicesSup-
- 16 ported message where the Total number of Services (see Table 2) is set equal to 0.

#### 3.3 StartService

The StartService message is used from the CDB Sink endpoint to signal to the CDB Source endpoint to start a specific service and commence putting data packets related to that data service on to the common data bus. The following table specifies the format of the message.

# bytes	Туре	Value	Description
2	U16	0xB103	Message-type
2	U16	4	Payload length
2	U16		Service Id
1	U8		Major version of service
1	U8		Minor version of service

Table 4: StartService Message

- The service version MUST be equal or smaller than the announced service version from the ServicesSup-
- 23 ported message. The CDB Source endpoint MUST respond with a ServiceResponse message with
- 24 the corresponding Service Id within 5s. Otherwise the CDB Source endpoint MUST respond with a Ser-
- 25 viceResponse message (Error Code set to 0x0010 Response pending) latest every 5s until the final
- 26 response message is available. The CDB Source endpoint MUST respond with a ServiceResponse mes-
- 27 sage (Error Code set to 0x0201 Launch failed) if no response can be provided within 2 min.
- In case the ServiceResponse messages are not received in time, the CDB Sink endpoint MUST consider
- 29 the launch finally failed and MUST send a StopService message.

## 3.4 StopService

1

5

15

- 2 The StopService message is used from the CDB Sink endpoint to signal to the CDB Source endpoint to
- 3 stop a specific service and stop putting data packets related to that service on to the common data bus. The
- 4 following table specifies the format of the message:

# bytes	Туре	Value	Description
2	U16	0xB104	Message-type
2	U16	2	Payload length
2	U16		Service Id

Table 5: StopService Message

- 6 The CDB Source endpoint MUST respond with a ServiceResponse message with the corresponding
- 7 Service Id within 5s.
- 8 In case the ServiceResponse message is not received in time, the CDB Sink endpoint MUST consider
- 9 the termination finally failed. No further action is required.

## 10 3.5 ServicePayload

- 11 The ServicePayload message delivers data from one CDB endpoint to the other one for any of the ser-
- 12 vices, which have been started. The payload can have any data format of its own depending on the available
- 13 services. The specification of the service payload is done within separate specifications and is out of the scope
- of this specification. The following table specifies the format of this message:

# bytes	Туре	Value	Description
2	U16	0xB105	Message-type
2	U16	3+M	Payload length
2	U16		Service Id
1	U8		Service Access Control
М	Array of	U8	Payload for the Service

- Table 6: ServicePayload Message
- 16 The Service Access Control byte of the ServicePayload message MUST be identical to the Service
- 17 Access Control byte from the respective service in the ServicesSupported message.
- 18 In case the Service Encryption bit is set to '1' in the Service Configuration bit of the Service listing, the entire
- 19 Service Access Control byte and the Service Payload MUST be encrypted. Each CDB endpoint MUST use
- 20 its allocated encryption mechanism and session key. The definition of the encryption mechanism and session
- 21 key are outside the scope of this specification.
- 22 The service payload length M, as defined in Table 6, MUST be equal or smaller than 8,100 bytes. Both
- 23 MirrorLink client and server MUST be able to receive a service payload up to this size.
- 24 Any service using the Common Data Bus, MAY split long payloads into smaller service payloads, fulfilling
- 25 the service payload length requirement. In such case, it is up to each service to fragment and re-assemble
- those longer packets.
- 27 The CDB Sink endpoint MUST respond with a ServiceResponse message with the corresponding Ser-
- 28 vice Id, if the ServicePayload message's Service Id is unknown or if the Service Configuration bits have
- 29 changed.

# 3.6 ServiceResponse

1

5

- 2 The ServiceResponse message MUST be used from the CDB Source or Sink endpoint to respond to a
- 3 message send from the CDB Sink or Source endpoint respectively (refer to Table 12 for details). The follow-
- 4 ing table specifies the format of this message:

# bytes	Туре	Value	Description
2	U16	0xB107	Message-type
2	U16	4	Payload length
2	U16		Service Id
2	U16		Response Value

Table 7: ServiceResponse Message

6 The error types are defined in the following table.

Response Value		Description
0x0001	Ok – Service started	Service has been successfully started.
0x0002	Ok – Service stopped	Service has been successfully terminated.
0x0010	Response pending	Response pending for another 5s
0x0101	Warning – Service run- ning	Service Id used in a StartService message refers to an already running service.
0x0102	Warning – Service not running	Service Id used in a StopService or ServicePayload message refers to a not-running service.
0x0201	Error – Service Launch failed	Service cannot be launched CDB Source endpoint MUST delist the service and send an updated ServicesSupported message.
0x0202	Error – Service Termination failed	Service cannot be terminated CDB Sink endpoint MUST ignore ServicePayloads from the service in question. CDB Sink endpoint MUST terminate the CDB.
0x0203	Error – Service reset	Service is reset from CDB Source endpoint. Service can be immediately started again using StartService message.
0x0204	Error – Service terminated	Service is terminated from CDB Source endpoint Service is (temporarily) unavailable. CDB Source end- point MUST send an updated ServicesSupported message.
0x0205	Error – Unknown Service Id	Service Id used in a ServicePayload, StartService or StopService message does not exist.
0x0206	Error – Wrong Access Control	Access Control in ServicePayload is wrong. CDB sink endpoint MUST stop the service.
0x0207	Error – Resource busy	Resource constraint service not started. Service resources are busy.
0x0208	Error – Wrong Sequence Number	Sequence number in PingResponse is wrong. Receiving CDB endpoint MUST terminate CDB.
0x0209	Error – Wrong CDB version number	Wrong version number in the ServicesRequested or ServicesReported message. CDB MUST be terminated.
0x020A	Error – Wrong Service version number	Invalid version number in the StartService message. Service is not started.
0x020B	Error – Unsupported Payload Format	The payload format used in the ServicePayload message is not supported from the Data Service.

Response Value	Description
	CDB sink endpoint MUST stop the service.

Table 8: Error Types for ServiceResponse message

## **2 3.7 ByeBye**

1

5

3 This message MUST be used from either the CDB Source or Sink endpoint to terminate the Common Data

4 Bus endpoint. The following table specifies the format of this message:

# bytes	Туре	Value	Description
2	U16	0xB109	Message-type
2	U16	0	Payload length

Table 9: BusError Message

6 Sending and receiving a ByeBye message MUST terminate the Common Data Bus, according chapter

7 2.2.1.2.

## 8 **3.8 Ping**

- 9 The Ping message MAY be used from either side, to check if the other side is still alive. The endpoint,
- 10 receiving a ping message, MUST reply with a PingResponse message containing the sequence number.
- 11 The following tables specify the format of the Ping message.

# bytes	Туре	Value	Description
2	U16	0xB108	Message-type
2	U16	2	Payload length
2	U16		Sequence number Arbitrary value for distinguishing PingResponse messages.

Table 10: Ping Message

- 13 If a Ping message is not replied via a PingResponse message within 10 seconds, the Common Data bus
- 14 MUST be considered non-functional. If either side recognizes the Common Data Bus to be non-functional, it
- MUST terminate the Common Data Bus as specified in chapter 2.2.1.2.

# 16 3.9 PingResponse

- 17 The PingResponse message MUST be used from either side, to respond to a Ping messages. The receiv-
- ing endpoint SHOULD send the PingResponse message, prior sending any other message. The following
- 19 tables specify the format of the PingResponse message.

# bytes	Туре	Value	Description
2	U16	0xB106	Message-type
2	U16 2		Payload length
2	U16		Sequence number in reply to a Ping message

Table 11: PingResponse Message

# 4 Message Flow

1

The following table summarizes the CDB Sink and Source Endpoint's message flow, it MUST follow. The sending endpoint is given in brackets in the Message Received column.

Message Received (from)	Message Responses	Comment
ServicesRequest	ServicesSupported	-
(Sink)	ServiceResponse (0x0209)	Wrong CDB version
ServicesSupported	No response	Wait for StartService
(Source)	ServiceResponse (0x0209)	Wrong CDB version
	ServiceResponse (0x0001)	Launch ok
	ServiceResponse (0x0010)	Response pending
	ServiceResponse (0x0101)	Launch failed – Already running
Start Service (Sink)	ServiceResponse (0x0201) ServicesSupported	Launch failed – Cannot launch service; update service list
,	ServiceResponse (0x0205)	Launch failed – Unknown Serviceld
	ServiceResponse (0x0207)	Launch failed – resource busy
	ServiceResponse (0x020A)	Launch failed – Invalid version
	ServiceResponse (0x0002)	Stop ok
Stop Service	ServiceResponse (0x0102)	Stop failed – Not running
(Sink)	ServiceResponse (0x0202)	Stop failed – Cannot stop service
,	ServiceResponse (0x0205)	Launch failed – Unknown Serviceld
	No response	Payload ok
	ServiceResponse (0x0101)	Payload failed – Service not running
	ServiceResponse (0x0203)	Payload failed – Service reset
ServicePayload	ServiceResponse (0x0204) ServicesSupported	Payload failed – Service terminated; update service list
(Sink)	ServiceResponse (0x0205)	Payload failed – Unknown Service Id
	ServiceResponse (0x0206)	Payload failed – Wrong Access Control
	ServiceResponse (0x020B)	Payload failed – Unsupported format
	No response	Payload ok
	ServiceResponse (0x0101)	Payload failed - Service not running
Comitos Doulond	ServiceResponse (0x0205)	Payload failed – Unknown Service Id
ServicePayload (Source)	ServiceResponse (0x0206) StopService	Payload failed – Wrong Access Control; stop service
	ServiceResponse (0x020B) StopService	Payload failed – Unsupported format; stop service
0 . 0	No action	Other response values
ServiceResponse	ByeBye	If response value = 0x0208
(Sink)	ByeBye	If response value = 0x0209
	No action	Other response values
	ByeBye	If response value = 0x0202
ServiceResponse	StartService or No action	If response value = 0x0203
(Source)	StopService	If response value = 0x0206
	ByeBye	If response value = 0x0208

Message Received (from)	Message Responses	Comment
	ByeBye	If response value = 0x0209
	StopService	If response value = 0x020B
ByeBye (Client Endpoint)	No response	Terminate the CDB session
ByeBye (Server Endpoint)	ByeBye	Terminate the CDB session
Ping (Sink / Source)	PingResponse	-
PingResponse	No response	Ping ok
(Sink / Source)	ServiceResponse (0x0208)	Ping failed – Wrong sequence number

Table 12: CDB Message Flow

## 5 REFERENCES

- 2 [1] Car Connectivity Consortium, "MirrorLink Application Server Service", Version 1.1; CCC-TS-024
- 4 [2] IETF, RFC 2119, Keys words for use in RFCs to Indicate Requirement Levels, March 1997. 5 http://www.ietf.org/rfc/rfc2119.txt

