Car Connectivity Consortium

MirrorLink®

Service Binary Protocol Test Specification

Version 1.1.3
(CCC-TS-019)

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	25 September 2014	Approved Errata Version

LIST OF CONTRIBUTORS

Brakensiek, Jörg	Microsoft Corporation
Hrabak, Robert (Editor)	General Motors
Park, Keun-Young	Nokia Corporation

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

TERMS AND ABBREVIATIONS

GPS	Global Positioning System

MirrorLink is a registered trademark of the Car Connectivity Consortium LLC.

UPnP is a registered trademark of UPnP Forum.

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

1 ABOUT

This document specifies test specification for MirrorLink Service Binary Protocol which is specified in MirrorLink Data Service Core specification [1].

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 [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 GENERAL TEST REQUIREMENTS

All packets and payloads shall be checked for integrity and validity of each field in every test case. Required data shall be specified within test cases as needed. Proper response latency of < 5 seconds shall bet tested in all case. Response-Continue packets MAY be exchanged as needed during any test case unless otherwise specified.

2.1 Execution of Test Cases

Every test case is uniquely identified by an identifier.

- A MirrorLink Server MUST pass all test cases, starting with SC, if it implements a CDB/SBP Source endpoint.
- A MirrorLink Server MUST pass all test cases, starting with SK, if it implements a CDB/SBP Sink endpoint
- A MirrorLink Client MUST pass all test cases, starting with SC, if it implements a CDB/SBP Source endpoint,
- A MirrorLink Client MUST pass all test cases, starting with SK, if it implements a CDB/SBP Sink endpoint

Every test case description includes an entry, whether the test cases is considered mandatory or not.

- Test cases marked as MANDATORY, MUST be executed.
- Test cases marked as CONDITIONAL, MUST be executed if the given condition is met.
- Test cases marked as CONDITIONAL, MUST NOT be executed if the given condition is not met.
- Test cases marked as NONE, MUST NOT be executed

2.2 Test Case Dependencies

A MirrorLink Server and Client, implementing Service Binary Protocol functionality MUST execute all Common Data Bus test cases, related to the implemented SBP functionality, as given below:

- A MirrorLink Server MUST pass all Server related CDB test cases
- A MirrorLink Client MUST pass all Client related CDB test cases
- A MirrorLink Server and Client MUST pass all CDB Source test cases, if it implements a CDB/SBP Source endpoint
- A MirrorLink Server and Client MUST pass all CDB Sink test cases, if it implements a CDB/SBP Sink endpoint

3 SINK TEST CASES

3.1 SK/SBP/SUBSCRIBE

Requirement: SBP Sec. 3.10, 3.4.3 - A SBP Sink starts and subscribes to a supported service on an SBP Source

Requirement: CONDITIONAL

Condition: SBP Sink supports SUBSCRIBE

3.1.1 Preparation Steps

Step	*Name*	*Description*	*Expected result*
1	*Send CDB Service Request*	Query available services supported by SBP Source that are available for communication/subscription.	*The test passes if* • *CDB Sink Endpoint sends correct service request from CDB Source Endpoint* • *CDB Source Endpoint returns a list of supported services*
2	*Start Supported Service Command*	SBP Sink triggers a service that was declared as supported by the SBP Source. This is either done automatically or manually from the test engineer, as described in the PIXIT. It is expected that the SBP Sink at least supports one of the two specified CDB/SBP services. Any of those two services are sufficient for this test case.	*The test passes if* • SBP Source receives a StartService request for a supported service • *SBP Source responds OK*

3.1.2 Objective Steps

Step	*Name*	*Description*	*Expected result*
3	*Subscribe Command*	SBP Sink subscribes to any object within the service it triggered.	*The test passes if* • *SBP Source receives a subscribe request to a started service* • *SBP Source may receive get requests prior to the subscribe request.*

3.1.3 Post Steps

None

3.2 SK/SBP/GET

Requirement: SBP Specs. 3.4.1 - A SBP Sink requests object data from SBP Source

Requirement: CONDITIONAL

Condition: SBP Sink supports GET

3.2.1 Preparation Steps

Step	*Name*	*Description*	*Expected result*
1	*Initialize Supported SBP Service*	SBP Sink queries SBP Source for supported services, requests the activation of a target service if not already running, and subscribes to it	*The test passes if* • *SBP Source receives a subscribe request to a started service* • *SBP Source responds OK*
2	*Start Supported Service Command*	SBP Sink triggers a service that was declared as supported by the SBP Source. This is either done automatically or manually from the test engineer, as described in the PIXIT. It is expected that the SBP Sink at least supports one of the two specified CDB/SBP services. Any of those two services are sufficient for this test case.	*The test passes if* • SBP Source receives a StartService request for a supported service • *SBP Source responds OK*

3.2.2 Objective Steps

Step	*Name*	*Description*	*Expected result*
3	*Request supported object data from Service*	SBP Sink sends a Get command to the SBP Source for valid data supported in the service.	*The test passes if* • SBP Sink requests valid object data from SBP Source • *SBP Source responds with payload and OK or no more session error code.*

3.2.3 Post Steps

None

3.3 SK/SBP/AUTHENTICATE

A SBP Sink requests object data from SBP Source with successful handling of SBP Source authentication challenge.

Requirement: SBP Specs. 3.4.1 and 3.7

Requirement: MANDATORY

Condition: None

3.3.1 Preparation Steps

Step	*Name*	*Description*	*Expected result*
1	*Initialize Supported SBP Service*	SBP Sink queries SBP Source for supported services, requests the activation	*The test passes if*

		of a target service if not already running.	- *SBP Source receives valid message to start service* - *SBP Source starts OK*
2	*Start Supported Service Command*	SBP Sink triggers a service that was declared as supported by the SBP Source. This is either done automatically or manually from the test engineer, as described in the PIXIT. It is expected that the SBP Sink at least supports one of the two specified CDB/SBP services. Any of those two services are sufficient for this test case.	*The test passes if* - SBP Source receives a StartService request for a supported service - *SBP Source responds OK*
3	*Request supported object data from Service*	SBP Sink sends a Get or Subscribe command to the SBP Source for valid data supported in the service, but receives and handles an authentication challenge from the SBP Source.	*The test passes if* - SBP Sink requests valid object data from SBP Source - *SBP Source responds with an Authentication Challenge*

3.3.2 Objective Steps

Step	*Name*	*Description*	*Expected result*
4	*Correct Authentication Response*	SBP Sink responds with an error message.	*The test passes if* - *SBP Sink sends a correct authentication response with "Feature not supported" error code.*

3.3.3 Post Steps

None

3.4 SK/SBP/SET

Not testable, as none of the current specified data services support SET command.

3.5 SK/SBP/ALIVE

A SBP Sink checks alive status of SBP Source.

Requirement: SBP Specs. 3.4.5

Requirement: Conditional

Condition: SBP Sink supports AliveRequest

3.5.1 Preparation Steps

Step	*Name*	*Description*	*Expected result*

1	*Start Supported SBP Service*	SBP Sink queries SBP Source for supported services and requests the activation of a target service if not already running	*The test passes if* • *SBP Source receives a start request to a started service* • *SBP Source responds OK*

3.5.2 Objective Steps

Step	*Name*	*Description*	*Expected result*
2	*Request Alive object data from Service*	SBP Sink sends an event to verify that the SBP Source is still responsive, and SBP Source responds correctly. May need PIXIT to trigger Alive requests.	*The test passes if* • SBP Sink requests Alive event • *SBP Source sends Alive reply*

3.5.3 Post Steps

None

3.6 SK/SBP/CONTINUE

A SBP Source delays the response to the SBP Sink

Requirement: SBP Specs. 3.4

Requirement: MANDATORY

Condition: None

3.6.1 Preparation Steps

Step	*Name*	*Description*	*Expected result*
1	*Initialize Supported SBP Service*	SBP Sink queries SBP Source for supported services, requests the activation of a target service if not already running, and subscribes to it	*The test passes if* • *SBP Source receives a subscribe request to a started service* • *SBP Source responds OK*
2	*Start Supported Service Command*	SBP Sink triggers a service that was declared as supported by the SBP Source. This is either done automatically or manually from the test engineer, as described in the PIXIT. It is expected that the SBP Sink at least supports one of the two specified CDB/SBP services. Any of those two services are sufficient for this test case.	*The test passes if* • SBP Source receives a StartService request for a supported service • *SBP Source responds OK*

3.6.2 Objective Steps

Step	*Name*	*Description*	*Expected result*

3	*Delayed response on SBP Sink commands*	SBP Sink sends a SET, GET or SUBSCRIBE commands to the SBP Source for valid data supported in the service and successfully assigns the value to the data object	*The test passes if* • SBP Source sends at least 2 Response-Continue message, prior completing the command. • SBP Sink does not stop the SBP Service

3.6.3 Post Steps

None

3.7 SK/SBP/ALIVE_NORESPONSE

A SBP Sink checks alive status of SBP Source and recognizes irrecoverable error, due to time out in Alive Response.

Requirement: SBP Specs. 3.4.5

Requirement: CONDITIONAL

Condition: SBP Sink supports AliveRequest

3.7.1 Preparation Steps

Step	*Name*	*Description*	*Expected result*
1	*Start Supported SBP Service*	SBP Sink queries SBP Source for supported services and requests the activation of a target service if not already running	*The test passes if* • *SBP Source receives a start request to a started service* • *SBP Source responds OK*
2	*Start Supported Service Command*	SBP Sink triggers a service that was declared as supported by the SBP Source. This is either done automatically or manually from the test engineer, as described in the PIXIT. It is expected that the SBP Sink at least supports one of the two specified CDB/SBP services. Any of those two services are sufficient for this test case.	*The test passes if* • SBP Source receives a StartService request for a supported service • *SBP Source responds OK*

3.7.2 Objective Steps

Step	*Name*	*Description*	*Expected result*
3	*Request Alive object data from Service*	SBP Sink sends an event to verify that the SBP Source is still responsive, and SBP Source does not reply .	*The test passes if* • SBP Sink requests Alive event • *SBP Source DOES NOT SEND Alive reply within appropriate timescale* • *CDB Sink sends a StopService message not earlier than 5s after sending the AliveRequest message*

3.7.3 Post Steps

Step	*Name*	*Description*	*Expected result*
4	*Clean-Up*	SBP Sink detects irrecoverable error and ends SBP service session	*The test passes if* • SBP Sink successfully ends CDB session

3.8 SK/SBP/ERROR_UnknownDataType

A SBP Sink receives a response with unknown data type.

Requirement: SBP Specs. 3.6.1.1

Requirement: Mandatory

Condition: None

3.8.1 Preparation Steps

Step	*Name*	*Description*	*Expected result*
1	*Initialize Supported SBP Service*	SBP Sink queries SBP Source for supported services, requests the activation of a target service if not already running, and subscribes to it	*The test passes if* • *SBP Source receives a subscribe request to a started service* • *SBP Source responds OK*
2	*Start Supported Service Command*	SBP Sink triggers a service that was declared as supported by the SBP Source. This is either done automatically or manually from the test engineer, as described in the PIXIT. It is expected that the SBP Sink at least supports one of the two specified CDB/SBP services. Any of those two services are sufficient for this test case.	*The test passes if* • SBP Source receives a StartService request for a supported service • *SBP Source responds OK*
3	*Send command to SBP Serer*	SBP Sink sends a GET/SUBSCRIBE command to SBP Source. SBP Source responds with an unknown data type.	*The test passes if* • *SBP Sink sends GET/SUBSCRIBE command*

3.8.2 Objective Steps

Step	*Name*	*Description*	*Expected result*
4	*Irrecoverable Error Detection*	SBP Sink recognizes invalid data type and ends SBP session	*The test passes if* • *CDB Sink sends a StopService message*

3.8.3 Post Steps

None

3.9 SK/SBP/ERROR_WrongEnd

A SBP Sink receives a response with invalid END_C.

Requirement: SBP Specs. 3.6.1.2

Requirement: Mandatory

Condition: None

3.9.1 Preparation Steps

Step	*Name*	*Description*	*Expected result*
1	*Initialize Supported SBP Service*	SBP Sink queries SBP Source for supported services, requests the activation of a target service if not already running, and subscribes to it	*The test passes if* • *SBP Source receives a subscribe request to a started service* • *SBP Source responds OK*
2	*Start Supported Service Command*	SBP Sink triggers a service that was declared as supported by the SBP Source. This is either done automatically or manually from the test engineer, as described in the PIXIT. It is expected that the SBP Sink at least supports one of the two specified CDB/SBP services. Any of those two services are sufficient for this test case.	*The test passes if* • SBP Source receives a StartService request for a supported service • *SBP Source responds OK*
3	*Send command to SBP Serer*	SBP Sink sends a valid command to SBP Source. SBP Source responds with an invalide END_C.	*The test passes if* • *SBP Sink sends SET/GET/SUBSCRIBE command*

3.9.2 Objective Steps

Step	*Name*	*Description*	*Expected result*
4	*Irrecoverable Error Detection*	SBP Sink recognizes invalid END_C and ends SBP session	*The test passes if* • *CDB Sink sends a StopService message*

3.9.3 Post Steps

None

3.10 SK/SBP/ERROR_UnknownUID

A SBP Sink receives a response containing an object with unknown Object UID.

Requirement: SBP Specs. 3.6.2.1

Requirement: Mandatory

Condition: None

3.10.1 Preparation Steps

Step	*Name*	*Description*	*Expected result*
1	*Initialize Supported SBP Service*	SBP Sink queries SBP Source for supported services, requests the activation of a target service if not already running, and subscribes to it	*The test passes if* - *SBP Source receives a subscribe request to a started service* - *SBP Source responds OK*
2	*Start Supported Service Command*	SBP Sink triggers a service that was declared as supported by the SBP Source. This is either done automatically or manually from the test engineer, as described in the PIXIT. It is expected that the SBP Sink at least supports one of the two specified CDB/SBP services. Any of those two services are sufficient for this test case.	*The test passes if* - SBP Source receives a StartService request for a supported service - *SBP Source responds OK*
3	*Send command to SBP Serer*	SBP Sink sends a valid command to SBP Source. SBP Source responds with an unknown object UID.	*The test passes if* - *SBP Sink sends SET/GET/SUBSCRIBE command*

3.10.2 Objective Steps

Step	*Name*	*Description*	*Expected result*
4	*Recoverable Error Detection*	SBP Sink recognizes unknown object UID and ignores it. SBP Source sends response with correct object UID	*The test passes if* - *SBP Sink continues without stopping the service.*

3.10.3 Post Steps

None

4 Source Test Cases

4.1 SC/SBP/SUBSCRIBE

Requirement: SBP Sec. 3.4.3 - A SBP Source starts and allows subscription to a supported service requested by a SBP Sink

Requirement: Mandatory

Condition: None

4.1.1 Preparation Steps

Step	*Name*	*Description*	*Expected result*
1	*Start Supported Service*	*SBP Source successfully responds to a valid Service Start command*	*The test pass if* • *SBP Source responds with all available services* • *The service requested is started* • *The SBP Source responds OK to the StartService command*
2	*Start Supported Service*	*SBP Source successfully responds to a valid Service Start command* It is expected that the SBP Source at least supports one of the two specified CDB/SBP services. Any of those two services are sufficient for this test case.	*The test passes if* • *The service requested is started* • *The SBP Source responds OK to the StartService command*

4.1.2 Objective Step

Step	*Name*	*Description*	*Expected result*
3	*Accept Supported Service Subscription*	*SBP Source successfully responds to a valid Service Subscribe command* *Subscribe to all objects (regular intervals as allowed in service description)* *Subscribe to the following subscription types:* - Interval - On-Change - Automatic Note: Interval or On-change subscription may not be supported (PIXIT).	*The test passes if* • *SBP Source does returns OK response* • *The SBP Source sends data on the subscribed service to the SBP Sink* • *SBP Source MAY respond with "feature not supported"*
4	*Receive Data*	*SBP Source provides data for subscribed objects within time, if feature is supported* *May need PIXIT information, if On-Change/Automatic subscription requires specific action to force a change.*	*The test passes if* • *SBP Source provides data in time; for interval subscription, allow for an extension of +10% (at least 100 ms) or* • *Subscribe feature is not supported for object in question.*

4.1.3 Post Steps

None

4.2 SC/SBP/GET_SUBSCRIBED

Requirement: SBP Specs. 3.4.1 - A SBP Sink requests object data from SBP Source while subscribed to the same Object.

Requirement: Mandatory

Condition: None

4.2.1 Preparation Steps

Step	*Name*	*Description*	*Expected result*
1	*Initialize Supported SBP Service*	SBP Source responds with SBP Source for supported services, accepts the activation of a target service if not already running, and permits subscription to it	*The test passes if* • *SBP Source receives a subscribe request to a started service* • *SBP Source responds OK*
2	*Start Supported Service*	*SBP Source successfully responds to a valid Service Start command* It is expected that the SBP Source at least supports one of the two specified CDB/SBP services. Any of those two services are sufficient for this test case.	*The test passes if* • *The service requested is started* • *The SBP Source responds OK to the StartService command*
3	*Accept Supported Service Subscription*	*SBP Source successfully responds to a valid Service Subscribe command* *Subscribe to an object (automatic)*	*The test passes if* • *SBP Source does returns OK response* • *The SBP Source MAY send data on the subscribed service to the SBP Sink* • *SBP Source MAY respond with “feature not supported”*

4.2.2 Objective Steps

Step	*Name*	*Description*	*Expected result*
4	*Return supported object data from Service*	SBP Source delivers the valid object data requested by the SBP Sink	*The test passes if* • SBP Sink requests valid object data from SBP Source for the same object, it just subscribed to. • *SBP Source responds with payload and OK or valid error code* • *Packet_id corresponds to GET command*

4.2.3 Post Steps

None

4.3 SC/SBP/GET_UNSUBSCRIBED

A SBP Sink requests object data from SBP Source when not subscribed to the SBP Object

Requirement: SBP Specs. 3.4.1

Requirement: Mandatory

Condition: None

4.3.1 Preparation Steps

Step	*Name*	*Description*	*Expected result*
1	*Start Supported SBP Service*	SBP Source responds to query with SBP Source supported services, accepts the activation of a target service if not already running	*The test passes if* • *SBP Source receives a start request to a supported service* • *SBP Source responds OK* • *SBP Sink does NOT subscribe to service*
2	*Start Supported Service*	*SBP Source successfully responds to a valid Service Start command* It is expected that the SBP Source at least supports one of the two specified CDB/SBP services. Any of those two services are sufficient for this test case.	*The test passes if* • *The service requested is started* • *The SBP Source responds OK to the StartService command*

4.3.2 Objective Steps

Step	*Name*	*Description*	*Expected result*
3	*Request supported object data from Service*	SBP Source delivers the valid object data requested by the SBP Sink	*The test passes if* • SBP Sink requests valid object data from SBP Source • *SBP Source responds with payload and OK*

4.3.3 Post Steps

None

4.4 SC/SBP/SET_SUBSCRIBE

Not testable, as none of the current specified data services support SET command.

4.5 SC/SBP/SET_UNSUBSCRIBED

Not testable, as none of the current specified data services support SET command.

4.6 SC/SBP/ERROR_NotWritable

A SBP Source receives SET command for non-writable object.

Requirement: SBP Specs. 3.4.2

Requirement: MANDATORY

Condition: None

4.6.1 Preparation Steps

Step	*Name*	*Description*	*Expected result*
1	*Initialize Supported SBP Service*	SBP Source responds with SBP Source for supported services, accepts the activation of a target service if not already running, and permits subscription to it	*The test passes if* • *SBP Source receives a subscribe request to a started service* • *SBP Source responds OK*
2	*Start Supported Service*	*SBP Source successfully responds to a valid Service Start command* It is expected that the SBP Source at least supports one of the two specified CDB/SBP services. Any of those two services are sufficient for this test case.	*The test passes if* • *The service requested is started* • *The SBP Source responds OK to the StartService command*

4.6.2 Objective Steps

Step	*Name*	*Description*	*Expected result*
3	*Try to set object data on Service that is not writable*	SBP Source recognizes Set command from the SBP Sink for supported service data and rejects it, as the data is not writable. This is true for GPS and Location objects.	*The test passes if* • SBP Sink sends object data to SBP Source • *SBP Source sends response with "Write not allowed" error code*

4.6.3 Post Steps

None

4.7 SC/SBP/ALIVE

A SBP Source responds to Alive Request.

Requirement: SBP Specs. 3.4.5

Requirement: Mandatory

Condition: None

4.7.1 Preparation Steps

Step	*Name*	*Description*	*Expected result*

1	*Initialize Supported SBP Service*	SBP Source responds with SBP Source for supported services, accepts the activation of a target service if not already running, and permits subscription to it	*The test passes if* • *SBP Source receives a subscribe request to a started service* • *SBP Source responds OK*
2	*Start Supported Service*	*SBP Source successfully responds to a valid Service Start command* It is expected that the SBP Source at least supports one of the two specified CDB/SBP services. Any of those two services are sufficient for this test case.	*The test passes if* • *The service requested is started* • *The SBP Source responds OK to the StartService command*

4.7.2 Objective Steps

Step	*Name*	*Description*	*Expected result*
3	*Request Alive object data from Service*	SBP Source receives an event to verify that it is still responsive, and responds correctly	*The test passes if* • SBP Sink requests Alive event • *SBP Source sends Alive reply*

4.7.3 Post Steps

None

4.8 SC/SBP/ERROR_WrongEnd

A SBP Source checks validity of END_C element and recognizes failure mode.

Requirement: SBP Specs. 3.6.1.2

Requirement: Mandatory

Condition: None

4.8.1 Preparation Steps

Step	*Name*	*Description*	*Expected result*
1	*Start Supported SBP Service*	SBP Source responds with SBP Source for supported services, accepts the activation of a target service if not already running, and permits subscription to it	*The test passes if* • *SBP Source receives a subscribe request to a started service* • *SBP Source responds OK*
2	*Start Supported Service*	*SBP Source successfully responds to a valid Service Start command* It is expected that the SBP Source at least supports one of the two specified CDB/SBP services. Any of those two services are sufficient for this test case.	*The test passes if* • *The service requested is started* • *The SBP Source responds OK to the StartService command*

4.8.2 Objective Steps

Step	*Name*	*Description*	*Expected result*
3	*Request Get of object data from started Service with invalid END_C*	SBP Source detects an invalid END_C in the SBP Sink's request and responds with associated error	*The test passes if* • SBP Source receives Get request • SBP Source recognizes Invalid END_C • *Send CDB Service Response message with error code "Service Reset"*

4.8.3 Post Steps

None

4.9 SC/SBP/CANCEL_SUBSCRIBE

A SBP Source correctly handles request to cancel a command.

Requirement: SBP Specs. 3.4.3

Requirement: Mandatory

Condition: None

4.9.1 Preparation Steps

Step	*Name*	*Description*	*Expected result*
1	*Start Supported SBP Service*	SBP Source responds to query with SBP Source supported services, accepts the activation of a target service if not already running	*The test passes if* • *SBP Source receives a start request to a supported service* • *SBP Source responds OK* • *SBP Sink does NOT subscribe to service*
2	*Start Supported Service*	*SBP Source successfully responds to a valid Service Start command* It is expected that the SBP Source at least supports one of the two specified CDB/SBP services. Any of those two services are sufficient for this test case.	*The test passes if* • *The service requested is started* • *The SBP Source responds OK to the StartService command*
3	*Receive Subscribe object data from Service*	*SBP Source successfully responds to a valid Service Subscribe command* *Subscribe to objects (regular intervals) as allowed in service description)*	*The test passes if* • *SBP Source does returns OK response* • *The SBP Source sends data on the subscribed service to the SBP Sink* • *SBP Source MAY respond with "feature not supported"*
4	*Receive Data*	*SBP Source provides data for subscribed objects within time, if feature is supported*	*The test passes if* • *SBP Source provides data in time or*

			• *Subscribe feature is not supported for object in question.*

4.9.2 Objective Steps

Step	*Name*	*Description*	*Expected result*
5	*Receive Cancel request from Service*	SBP Source receives a Cancel request for subscription from SBP Sink, then	*The test passes if* • SBP Source receives Cancel request • *SBP Source responds with OK*

4.9.3 Post Steps

Step	*Name*	*Description*	*Expected result*
6	*Object Data Change on Source*	SBP Source changes object data of service of former subscription	*The test passes if* • SBP Source DOES NOT send a data update to the SBP Sink

4.10 SC/SBP/ERROR_UnknownObjectUID

A SBP Source correctly responds to a command for member variables not supported within service

Requirement: SBP Specs. 3.6.2.2 and 3.6.4

Requirement: Mandatory

Condition: None

4.10.1 Preparation Steps

Step	*Name*	*Description*	*Expected result*
1	*Initialize Supported SBP Service*	SBP Source responds with SBP Source for supported services, accepts the activation of a target service if not already running, and permits subscription to it	*The test passes if* • *SBP Source receives a subscribe request to a started service* • *SBP Source responds OK*
2	*Start Supported Service*	*SBP Source successfully responds to a valid Service Start command* It is expected that the SBP Source at least supports one of the two specified CDB/SBP services. Any of those two services are sufficient for this test case.	*The test passes if* • *The service requested is started* • *The SBP Source responds OK to the StartService command*

4.10.2 Objective Steps

Step	*Name*	*Description*	*Expected result*
3	*Receive Get for unsupported object*	SBP Source receives a GET command from SBP Sink with a recoverable error with appropriate error code	*The test passes if* • SBP Source recognizes error

			- *SBP Source sends response with error code of "Unknown UID"*

4.10.3 Post Steps

Step	*Name*	*Description*	*Expected result*
4	*Command retry*	SBP Source receives a valid command from SBP Sink and responds correctly	*The test passes if* - SBP Sink successfully responds to the valid version of the command

4.11 SC/SBP/ERROR_UnknownCommand

A SBP Source correctly responds to an unknwon command for member variables not supported within service

Requirement: SBP Specs. 3.6.2.2 and 3.6.4

Requirement: Mandatory

Condition: None

4.11.1 Preparation Steps

Step	*Name*	*Description*	*Expected result*
1	*Initialize Supported SBP Service*	SBP Source responds with SBP Source for supported services, accepts the activation of a target service if not already running, and permits subscription to it	*The test passes if* - *SBP Source receives a subscribe request to a started service* - *SBP Source responds OK*
2	*Start Supported Service*	*SBP Source successfully responds to a valid Service Start command* It is expected that the SBP Source at least supports one of the two specified CDB/SBP services. Any of those two services are sufficient for this test case.	*The test passes if* - *The service requested is started* - *The SBP Source responds OK to the StartService command*

4.11.2 Objective Steps

Step	*Name*	*Description*	*Expected result*
3	*Receive unknown command*	SBP Source receives a unknown command from SBP Sink	*The test passes if* - SBP Source recognizes error - *SBP Source sends response with error code of "Unknown Command"*

4.11.3 Post Steps

Step	*Name*	*Description*	*Expected result*

4	*Command retry*	SBP Source receives a valid command from SBP Sink and responds correctly	*The test passes if* • SBP Sink successfully responds to the valid version of the command

4.12 SC/SBP/ERROR_WrongSubscriptionType

A SBP Sink sends a command with an invalid Object UID and SBP Source recognizes and returns proper error.

Requirement: SBP Specs. 3.6.2.1 and 3.6.4

Requirement: Mandatory

Condition: None

4.12.1 Preparation Steps

Step	*Name*	*Description*	*Expected result*
1	*Initialize Supported SBP Service*	SBP Source responds with SBP Source for supported services, accepts the activation of a target service if not already running, and permits subscription to it	*The test passes if* • *SBP Source receives a subscribe request to a started service* • *SBP Source responds OK*
2	*Start Supported Service*	*SBP Source successfully responds to a valid Service Start command* It is expected that the SBP Source at least supports one of the two specified CDB/SBP services. Any of those two services are sufficient for this test case.	*The test passes if* • *The service requested is started* • *The SBP Source responds OK to the StartService command*

4.12.2 Objective Steps

Step	*Name*	*Description*	*Expected result*
3	*Receive SUBSCRIBE with unsupported subscription type*	SBP Source receives a SUBSCRIBE command from SBP Sink with unknown subscription type	*The test passes if* • SBP Source recognizes error • *SBP Source sends response with error code of ”Wrong subscription type”*

4.12.3 Post Steps

Step	*Name*	*Description*	*Expected result*
4	*Command retry*	SBP Source receives a valid command from SBP Sink and responds correctly	*The test passes if* • SBP Sink successfully responds to the valid version of the command

4.13 SC/SBP/ERROR_CommandNotPending

A SBP Source responds to a cancel command for a non-subscribed object.

Requirement: SBP Specs. 3.6.3.1

Requirement: Mandatory

Condition: None

4.13.1 Preparation Steps

Step	*Name*	*Description*	*Expected result*
1	*Initialize Supported SBP Service*	SBP Source responds with SBP Source for supported services, accepts the activation of a target service if not already running, and permits subscription to it	*The test passes if* • *SBP Source receives a subscribe request to a started service* • *SBP Source responds OK*
2	*Start Supported Service*	*SBP Source successfully responds to a valid Service Start command* It is expected that the SBP Source at least supports one of the two specified CDB/SBP services. Any of those two services are sufficient for this test case.	*The test passes if* • *The service requested is started* • *The SBP Source responds OK to the StartService command*

4.13.2 Objective Steps

Step	*Name*	*Description*	*Expected result*
3	*Cancel command*	SBP Source receives a cancel command from SBP Sink without a prior subscribe command	*The test passes if* • SBP Source recognizes error • *SBP Source responds "Command not pending"*

4.13.3 Post Steps

None

5 REFERENCES

[1] Car Connectivity Consortium, "MirrorLink 1.1 – Service Binary Protocol", Version 1.1; CCC-TS-018

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