
Car Connectivity Consortium

MirrorLink®

HSML Test Specification

Version 1.2.1
(CCC-TS-066)



Copyright © 2011-2015 Car Connectivity Consortium LLC
All rights reserved
Confidential

1 **VERSION HISTORY**

Version	Date	Comment
1.2.0	11 September 2014	Approved Version
1.2.1	30 April 2015	Approved Errata Version

3 **LIST OF CONTRIBUTORS**

Benesch, Matthias	Mercedes-Benz Research & Development North America
Brakensiek, Jörg	Microsoft Corporation
Chung, Bill	HTC Corp.
Ed, Pichon	CCC
Wei, Joe	HTC Corp.

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 PROPERTY RIGHTS 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.

Copyright © 2015. CCC LLC.

TABLE OF CONTENTS

VERSION HISTORY	2
LIST OF CONTRIBUTORS	2
LEGAL NOTICE	3
TABLE OF CONTENTS	4
TERMS AND ABBREVIATIONS	6
1 ABOUT	7
2 DEFINITIONS	8
2.1 EXECUTION OF TEST CASES	8
2.2 SERVER DEFINITION	8
2.2.1 HSML Source Launch	8
2.2.2 HSML Source Configuration	8
2.2.3 HSML Source Start Operation	9
2.2.4 Intentional HSML Source Clean Up	9
2.2.5 Unintentional HSML Source Clean Up	10
2.3 CLIENT DEFINITION	10
2.3.1 HSML Sink Launch	10
2.3.2 HSML Sink Configuration	10
2.3.3 HSML Sink Start Operation	11
2.3.4 Intentional HSML Sink Clean Up	12
2.3.5 Unintentional HSML Sink Clean Up	12
3 SERVER FEATURE TEST CASES	13
3.1 HSML SOURCE AT MIRRORLINK SERVER	13
3.1.1 SR/HSML/USB/MultiplePersonalities	13
3.1.2 SR/HSML/USB/Endpoint	13
3.1.3 SR/HSML/USB/ShortPacket	14
3.2 HSML INITIALIZATION PHASE	15
3.2.1 SR/HSML/INIT/HsmlPseudoEncoding	15
3.2.2 SR/HSML/INIT/Identification	16
3.2.3 SR/HSML/INIT/Version	16
3.2.4 SR/HSML/INIT/GetParameters	16
3.2.5 SR/HSML/INIT/SetParameters	17
3.3 HSML BASIC OPERATION	18
3.3.1 SR/HSML/BASIC/NonIncrementalFbUpdate	18
3.3.2 SR/HSML/BASIC/IncrementalFbUpdate	19
3.3.3 SR/HSML/BASIC/StreamingFbUpdate	19
3.3.4 SR/HSML/BASIC/OnDemandFbUpdate	20
3.3.5 SR/HSML/BASIC/PauseStreamingFbUpdate	20
3.3.6 SR/HSML/BASIC/FbHeader	21
3.3.7 SR/HSML/BASIC/FrameRateAdjustment	22
3.3.8 SR/HSML/BASIC/PixelFormatChange	23
3.3.9 SR/HSML/BASIC/NonIncrementalContextInformation	23
3.3.10 SR/HSML/BASIC/IncrementalContextInformation	24
3.4 HSML PIXEL FORMAT SUPPORT	25
3.4.1 SR/HSML/PIXEL/ARGB888LittleEndianRAW	25
3.4.2 SR/HSML/PIXEL/ARGB888BigEndianRAW	25
3.4.3 SR/HSML/PIXEL/RGB565LittleEndianRAW	26
3.4.4 SR/HSML/PIXEL/RGB565BigEndianRAW	27
3.4.5 SR/HSML/PIXEL/ARGB888LittleEndianRLE	27
3.4.6 SR/HSML/PIXEL/ARGB888BigEndianRLE	28

1	3.4.7	SR/HSML/PIXEL/RGB565LittleEndianRLE.....	29
2	3.4.8	SR/HSML/PIXEL/RGB565BigEndianRLE.....	30
3	3.5	BLOCKING NOTIFICATIONS	31
4	3.5.1	SR/HSML/BLOCK/FbBlockingNotification.....	31
5	4	CLIENT FEATURE TEST CASES	32
6	4.1	HSML SINK AT MIRRORLINK CLIENT	32
7	4.1.1	CL/HSML/USB/MultiplePersonalities.....	32
8	4.1.2	CL/HSML/USB/ShortPacket.....	32
9	4.2	HSML INITIALIZATION PHASE.....	33
10	4.2.1	CL/HSML/INIT/HsmlPseudoEncoding.....	33
11	4.2.2	CL/HSML/INIT/Identification.....	33
12	4.2.3	CL/HSML/INIT/Version.....	34
13	4.2.4	CL/HSML/INIT/Parameters.....	34
14	4.3	HSML BASIC OPERATION	35
15	4.3.1	CL/HSML/BASIC/NonIncrementalFbUpdate.....	35
16	4.3.2	CL/HSML/BASIC/IncrementalFbUpdate.....	36
17	4.3.3	CL/HSML/BASIC/OnDemandFbUpdate	36
18	4.3.4	CL/HSML/BASIC/ContextInformation.....	37
19	4.3.5	CL/HSML/BASIC/ParallelVncData.....	37
20	4.4	HSML PIXEL FORMAT SUPPORT.....	38
21	4.4.1	CL/HSML/PIXEL/ARGB888RAW.....	38
22	4.4.2	CL/HSML/PIXEL/RGB565RAW.....	38
23	4.4.3	CL/HSML/PIXEL/ARGB888RLE.....	39
24	4.4.4	CL/HSML/PIXEL/RGB565RLE.....	39
25	4.5	BLOCKING NOTIFICATIONS	40
26	4.5.1	CL/HSML/BLOCK/FbBlocked.....	40
27	5	REFERENCES.....	41
28			

1 TERMS AND ABBREVIATIONS

2	HSML	High Speed Media Link
3	ML	MirrorLink
4	RFB	Remote Framebuffer
5	UPnP	Universal Plug and Play
6	USB	Universal Serial Bus
7	VNC	Virtual Network Computing

8
9 MirrorLink is a registered trademark of Car Connectivity Consortium LLC

10 RFB and VNC are registered trademarks of RealVNC Ltd.

11 UPnP is a registered trademark of UPnP Implementers Corporation.

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

1 ABOUT

This document specifies HSML connectivity conformance test cases for the HSML Specification [2].

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

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

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

2 DEFINITIONS

2.1 Execution of Test Cases

Every test case is uniquely identified by an identifier.

- A HSML Source MUST pass all test cases, starting with SR.
- A HSML Sink MUST pass all test cases, starting with CL.

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 Server Definition

The following definitions are frequently used in different server test cases. Usage is indicated by the given designator name.

2.2.1 HSML Source Launch

This definition contains all necessary steps, which does launch an HSML Source on the MirrorLink server. The HSML Source is waiting for the HSML Sink to connect via USB vendor specific class communication.

Step	Name	Description	Expected Result
1	VNC Server Launch	See Definitions in [3]	
2	VNC Server Handshake	See Definitions in [3]	
3	VNC Server Configuration	See Definitions in [3] and the encoding type -527 MUST be included.	
4	Start HSML Framebuffer Update	Send a VNC Framebuffer Update Request message with X and Y position set to 0 and width and height set to the negotiated framebuffer resolution during the VNC session.	<ul style="list-style-type: none">• A valid VNC Framebuffer Update message arrives• Message contains HSML pseudo encoding• A valid UUID in the HSML pseudo encoding.

Table 1: HSML Source Launch – Test Steps

2.2.2 HSML Source Configuration

This definition contains all necessary steps to complete the HSML Source Configuration. It is based on successful completion of the HSML Source Launch.

Step	Name	Description	Expected Result
1	HSML Source Identification	Send a GetIdentifier request to every connected HSML device. Select one device whose UUID is identical to that in the HSML pseudo encoding to communicate with afterwards.	<ul style="list-style-type: none">• A valid UUIDv4, as defined in [4], receives• One of received UUIDs MUST be the same as the UUID in the HSML pseudo encoding

2	HSML Source Version Verification	Send a GetVersion request with wValue set to 0x0100	<ul style="list-style-type: none"> • A valid HSML version receives • The value MUST be 0x0100.
3	HSML Source Configuration	Send a GetParameters request.	<ul style="list-style-type: none"> • Valid Get Parameters reply arrive • ARGB888 and RGB565 MUST be supported • RAW encoding MUST be supported • Reserved bits MUST be all zeroes: bmCapabilities – Bit 2 to 31 bmPixelFormatSupported – Bit 6 to 31 bmEncodingSupported – bit 2 to 31
4	Set Parameters	Send a SetParameters request with following values set: bmCapabilities: All zeroes. bPixelFormat: 2 (RGB565) bPadding: All zeroes wPadding: All zeroes bmEncodingSupported: bit 0 (RAW) is set and all other bits are zeroes.	<ul style="list-style-type: none"> • No STALL PID receives • No other errors occurred in USB connection

Table 2: HSML Source Configuration – Test Steps

2.2.3 HSML Source Start Operation

This definition contains all necessary steps to start HSML framebuffer updates. It is based on successful completion of the HSML Source Configuration.

Step	Name	Description	Expected Result
1	Streaming Mode HSML Framebuffer Updates	Send HSML protocol Start-FramebufferTransmission request with Streaming Mode Feature Enable This step MUST be skipped if client enabled the On-Demand mode	<ul style="list-style-type: none"> • Valid framebuffer update events • Framebuffer update event follows the framebuffer header definition in [2] • Framebuffer update event is sent again automatically
2	On-Demand Mode HSML Framebuffer Updates	Continue to provide HSML protocol StartFramebufferTransmission request This step MUST be skipped if client enabled the Streaming mode	<ul style="list-style-type: none"> • Valid framebuffer update events • Framebuffer update event follows the framebuffer header definition in [2]

Table 3: HSML Source Operation – Test Steps

2.2.4 Intentional HSML Source Clean Up

This definition contains all necessary steps to intentionally terminate the HSML Source.

Step	Name	Description	Expected Result
1	Stop Streaming Mode	Send HSML protocol Stop-FramebufferTransmission from client	<ul style="list-style-type: none"> • HSML Source stops sending framebuffer event

		This step MUST be skipped if server is not at streaming mode	Note: The HSML message may be received by the CTS after the VNC ByeBye message.
2	Intentional VNC Server Clean Up	See definitions in [4].	

Table 4: Intentional HSML Source Clean Up – Test Steps

2.2.5 Unintentional HSML Source Clean Up

This definition contains all necessary steps to unintentionally terminate the HSML Source.

Step	Name	Description	Expected Result
1	VNC Byebye	Send VNC Byebye message	

Table 5: Unintentional HSML Source Clean Up – Test Steps

2.3 Client Definition

The following definitions are frequently used in different client test cases. Usage is indicated by the given designator name.

2.3.1 HSML Sink Launch

This definition contains all necessary steps, which does launch a HSML Sink on the MirrorLink with HSML capability client. The HSML Sink is making the connection to the HSML Source via USB vendor specific class communication.

Step	Name	Description	Expected Result
1	VNC Client Launch	See Definitions in [3].	
2	VNC Client Handshake	See Definitions in [3].	
3	VNC Client Configuration	See Definitions in [3].	<ul style="list-style-type: none"> Set Encodings includes MirrorLink pseudo encoding, Context Information pseudo encoding and HSML pseudo encoding.
4	VNC Full Framebuffer Update Request	Wait for the initial full VNC framebuffer update request	<ul style="list-style-type: none"> A VNC Framebuffer Update Request message receives Non-incremental Requested full framebuffer Requested framebuffer area within the server's framebuffer resolution as provided in VNC Server Init Requested framebuffer area within the client's max. supported resolution as provided through PIXIT information.

Table 6: HSML Sink Launch – Test Steps

2.3.2 HSML Sink Configuration

This definition contains all necessary steps to complete the HSML Sink Configuration. It is based on successful completion of the HSML Sink Launch.

Step	Name	Description	Expected Result
1	HSML Source Identifier Generation	Generate a UUID, as defined in [5]. Send a VNC Framebuffer Update with HSML pseudo encoding including the generated UUID.	<ul style="list-style-type: none"> A valid GetIdentifier request receives The wLength MUST be 16.
2	HSML Version Verification	Send the generated UUID in step 1 in response to the GetIdentifier request Wait for the GetVersion request	<ul style="list-style-type: none"> A valid GetVersion receives The wLength MUST be 2 The wValue is given the correct HSML version
3	HSML Source Configuration	Send the 0x0100 in response to the GetVersion request Wait for the GetParameters request	<ul style="list-style-type: none"> A valid GetParameters request receives The wLength MUST be 16
4	Set Parameters	Send the follow HSML parameters in response to GetParameters request: <ul style="list-style-type: none"> wWidth and wHeight set to 800 and 480 Pixel format ARGB888 and RGB565 Support only RAW encoding Little endian byte order used Not support the incremental update 	<ul style="list-style-type: none"> A valid SetParameters request receives The wLength MUST be 12 Non-incremental update is used Pixel format ARGB888 or RGB565 is selected Support RAW encoding.

Table 7: HSML Sink Configuration – Test Steps

2.3.3 HSML Sink Start Operation

This definition contains all necessary steps to start HSML framebuffer updates. It is based on successful completion of the HSML Sink Configuration.

Step	Name	Description	Expected Result
1	Streaming Mode HSML Framebuffer Updates	Receive HSML protocol Start-FramebufferTransmission request with Streaming Mode Feature Enable Send framebuffer updates continuously End each framebuffer update with a USB short packet This step MUST be skipped if client enabled the On-Demand mode	<ul style="list-style-type: none"> Streaming Mode enabled StartFramebufferTransmission request arrives The SetMaxFrameRate request MAY arrive. The HSML sink MUST set the maximum frame rate to 30 (if ever sent).
2	On-Demand Mode HSML Framebuffer Updates	Receive HSML protocol Start-FramebufferTransmission request with On-Demand Mode Feature Enable Send a framebuffer update This step MUST be skipped if client enabled the Streaming mode	<ul style="list-style-type: none"> On-Demand Mode enabled StartFramebufferTransmission request arrives

Table 8: HSML Sink Operation – Test Steps

2.3.4 Intentional HSML Sink Clean Up

This definition contains all necessary steps to intentionally terminate the HSML Sink.

Step	Name	Description	Expected Result
1	VNC ByeBye	Send VNC ByeBye message	
2	Stop Streaming Mode	Receive HSML protocol Stop-FramebufferTransmission request This step MUST be skipped if server is not at streaming mode	<ul style="list-style-type: none">• HSML client sends stop framebuffer transmission event <p>Note: The HSML message may be received by the CTS before the ByeBye message.</p>

Table 9: Intentional HSML Sink Clean Up – Test Steps

2.3.5 Unintentional HSML Sink Clean Up

This definition contains all necessary steps to unintentionally terminate the HSML Sink.

Step	Name	Description	Expected Result
1	VNC ByeBye	Send VNC ByeBye message	

Table 10: Unintentional HSML Sink Clean Up – Test Steps

3 SERVER FEATURE TEST CASES

3.1 HSML Source at MirrorLink Server

3.1.1 SR/HSML/USB/MultiplePersonalities

Requirement: MANDATORY

Condition: None

Test if the server device supports multiple USB personalities which include HSML and CDC/NCM.

Step	Name	Description	Expected Result
1	Physical USB cable connection.	Connect MirrorLink Server to MirrorLink Client via a USB cable	<ul style="list-style-type: none"> MirrorLink Server is recognized as a USB device
2	USB device identification	Test that the Server USB device has been identified by the Client.	<ul style="list-style-type: none"> USB host is able to read the USB device descriptor
3	CDC/NCM and HSML personalities selection	Make sure that CDC/NCM and HSML personalities are loaded in the Server	<ul style="list-style-type: none"> USB device has personalities including CDC/NCM and HSML CDC/NCM interface descriptor <ul style="list-style-type: none"> (bInterfaceClass: 0x02) (bInterfaceSubclass: 0x0D) HSML interface descriptor <ul style="list-style-type: none"> (bInterfaceClass: 0xFF) (bInterfaceSubclass: 0xCC) (bInterfaceProtocol: 0x01)
4	USB disconnection	Unplug the USB cable connecting Server and Client	

Table 11: Multiple USB Personalities Support

3.1.2 SR/HSML/USB/Endpoint

Requirement: MANDATORY

Condition: None

Test if the number of endpoints and their attributes are valid in the HSML interface and endpoint descriptors.

Step	Name	Description	Expected Result
1	Physical USB cable connection.	Connect MirrorLink Server to MirrorLink Client via a USB cable	<ul style="list-style-type: none"> MirrorLink Server is recognized as a USB device
2	USB device identification	Test that the Server USB device has been identified by the Client.	<ul style="list-style-type: none"> USB host is able to read the USB device descriptor
3	CDC/NCM and HSML personalities selection	Make sure that CDC/NCM and HSML personalities are loaded in the Server	<ul style="list-style-type: none"> USB device has personalities including CDC/NCM and HSML CDC/NCM interface descriptor

			<ul style="list-style-type: none"> ○ (bInterfaceClass: 0x02) ○ (bInterfaceSubclass: 0x0D) • HSML interface descriptor <ul style="list-style-type: none"> ○ (bInterfaceClass: 0xFF) ○ (bInterfaceSubclass: 0xCC) ○ (bInterfaceProtocol: 0x01)
4	Endpoint descriptor verification	Test that the number of endpoints in the HSML interface descriptor and the attributes of them are valid.	<ul style="list-style-type: none"> • USB Host can read and parse the Server USB descriptors • The bNumEndpoints field in HSML interface descriptor MUST be 1. • The endpoint MUST be BULK IN and its wMaxPacketSize MUST be 512 for USB 2.0 and 1024 for USB 3.0
4	USB disconnection	Unplug the USB cable connecting Server and Client	

Table 12: BULK IN Endpoint

3.1.3 SR/HSML/USB/ShortPacket

Requirement: MANDATORY

Condition: None

Test if the HSML source ends a framebuffer with a short packet.

Step	Name	Description	Expected Result
1	HSML Source Launch	<p>See Definitions</p> <p>Indicate no down-scaling support in Client Display Configuration</p> <p>Set Client Framebuffer resolution in Client Display Configuration to the following values (in separate runs of the test case)</p> <ul style="list-style-type: none"> • Width * Height * 2 is the multiple of 512 for USB 2.0 or 1024 for USB 3.0 and the ratio of Width and Height must be identical to the MirrorLink server's framebuffer aspect ratio and must be within the MirrorLink server's framebuffer resolution • Width * Height * 2 is NOT the multiple of 512 for USB 2.0 or 1024 for USB 3.0 and the ratio of Width and Height must be identical to the MirrorLink server's framebuffer aspect ratio and must be within the MirrorLink server's framebuffer resolution 	

2	HSML Source Configuration	See Definitions	
3	Streaming Mode HSML Framebuffer Updates	Send a StartFramebufferTransmission request with the streaming mode enabled.	<ul style="list-style-type: none"> • Valid framebuffer update events • Framebuffer update event follows the framebuffer header definition in [2] • Framebuffer update event is sent again automatically. • Framebuffer content is replicating server's framebuffer • A zero length packet is received for each Framebuffer update event, if the framebuffer size is multiple of 512 for USB 2.0 or 1024 for USB 3.0 • A USB short packet is received for each Framebuffer update event, if the framebuffer size is NOT multiple of 512 for USB 2.0 or 1024 for USB 3.0
4	Intentional HSML Source Clean Up	See Definitions	

Table 13: Short Packet

3.2 HSML Initialization Phase

3.2.1 SR/HSML/INIT/HsmlPseudoEncoding

Requirement: MANDATORY

Condition: None

Test if the HSML Source includes the HSML pseudo encoding (-527) in its VNC Framebuffer Update message in response to a VNC Framebuffer Update Request message.

Step	Name	Description	Expected Result
1	VNC Server Launch	See Definitions in [3]	
2	VNC Server Handshake	See Definitions in [3]	
3	VNC Server Configuration	See Definitions in [3] and the encoding type -527 MUST be included.	
4	Start HSML Framebuffer update	Send a VNC Framebuffer Update Request message with X and Y position set to 0 and width and height set to the negotiated framebuffer resolution during the VNC session.	<ul style="list-style-type: none"> • A valid VNC Framebuffer Update message arrives • Message contains HSML pseudo encodings.

5	Intentional HSML Source Clean Up	See Definitions.	
---	--	------------------	--

Table 14: HSML Pseudo Encoding

3.2.2 SR/HSML/INIT/Identification

Requirement: MANDATORY

Condition: None

Test if the HSML Source sends the same UUID in both the returned value of GetIdentifier request and the UUID filed in the HSML pseudo encoding.

Step	Name	Description	Expected Result
1	HSML Source Launch	See Definitions	
2	HSML Source Identification	Send a GetIdentifier request	<ul style="list-style-type: none"> A valid UUIDv4, as defined in [4], receives The received UUID MUST be the same as the UUID in the HSML pseudo encoding.
3	Intentional HSML Source Clean Up	See Definitions.	

Table 15: HSML Source Identification

3.2.3 SR/HSML/INIT/Version

Requirement: MANDATORY

Condition: None

Test if the HSML Source sends the correct HSML version in response to the GetVersion request.

Step	Name	Description	Expected Result
1	HSML Source Launch	See Definitions	
2	HSML Source Identification	Send a GetIdentifier request	<ul style="list-style-type: none"> A valid UUIDv4, as defined in [4], receives The received UUID MUST be the same as the UUID in the HSML pseudo encoding.
3	HSML Source Version Verification	Send a GetVersion request with wValue set to 0x0100	<ul style="list-style-type: none"> A valid HSML version receives The version MUST be 0x0100
4	Intentional HSML Source Clean Up	See Definitions.	

Table 16: HSML Version

3.2.4 SR/HSML/INIT/GetParameters

Requirement: MANDATORY

- 1 Condition: None
- 2 Test if the HSML Source returns the valid parameters in response to the GetParameters request.

Step	Name	Description	Expected Result
1	HSML Source Launch	See Definitions	
2	HSML Source Identification	Send a GetIdentifier request	<ul style="list-style-type: none"> • A valid UUIDv4, as defined in [4], receives • The received UUID MUST be the same as the UUID in the HSML pseudo encoding.
3	HSML Source Version Verification	Send a GetVersion request with wValue set to 0x0100	<ul style="list-style-type: none"> • A valid HSML version receives • The version MUST be 0x0100
4	HSML Source Configuration	Send a GetParameters request.	<ul style="list-style-type: none"> • Valid Get Parameters reply arrive • ARGB 888 and RGB 565 MUST be supported • RAW encoding MUST be supported • Reserved bits MUST be all zeroes: bmCapabilities – Bit 2 to 31 bmPixelFormatSupported – Bit 6 to 31 bmEncodingSupported – bit 2 to 31
5	Intentional HSML Source Clean Up	See Definitions.	

Table 17: GetParameters Request

3.2.5 SR/HSML/INIT/SetParameters

- Requirement: MANDATORY
- Condition: None
- 7 Test if the HSML Source returns the valid parameters in response to the GetParameters request.

Step	Name	Description	Expected Result
1	HSML Source Launch	See Definitions	
2	HSML Source Identification	Send a GetIdentifier request	<ul style="list-style-type: none"> • A valid UUIDv4, as defined in [5], receives • The received UUID MUST be the same as the UUID in the HSML pseudo encoding.
3	HSML Source Version Verification	Send a GetVersion request with wValue set to 0x0100	<ul style="list-style-type: none"> • A valid HSML version receives • The version MUST be 0x0100

4	HSML Source Configuration	Send a GetParameters request.	<ul style="list-style-type: none"> Valid Get Parameters reply arrive ARGB888 and RGB565 MUST be supported RAW data MUST be supported Reserved bits MUST be all zeroes: bmCapabilities – Bit 2 to 31 bmPixelFormatSupported – Bit 6 to 31 bmEncodingSupported – bit 2 to 31
5	Set Parameters	Send a SetParameters request with following values set: bmCapabilities: All zeroes. bPixelFormat: 2 (RGB565) bPadding: All zeroes wPadding: All zeroes bmEncodingSupported: bit 0 (RAW) is set and all other bits are zeroes.	<ul style="list-style-type: none"> No STALL PID receives No other errors occurred in USB connection
6	Intentional HSML Source Clean Up	See Definitions.	

Table 18: SetParameters Request

3.3 HSML Basic Operation

3.3.1 SR/HSML/BASIC/NonIncrementalFbUpdate

Requirement: MANDATORY

Condition: None

HSML Source MUST send framebuffer update event continuously even its display content doesn't change after receiving StartFramebufferTransmission request with Streaming Mode enabled and the FBUpdateOnChange bit in the SetParameters request doesn't set.

Step	Name	Description	Expected Result
1	HSML Source Launch	See Definitions	
2	HSML Source Configuration	See Definitions	
3	Streaming Mode HSML Framebuffer Updates	Send a StartFramebufferTransmission request with the streaming mode enabled.	<ul style="list-style-type: none"> Valid framebuffer update events Framebuffer update event follows the framebuffer header definition in [2] Framebuffer update event is sent again automatically
4	Intentional HSML Source Clean Up	See Definitions	

Table 19: Non-Incremental Framebuffer Update

3.3.2 SR/HSML/BASIC/IncrementalFbUpdate

Requirement: CONDITIONAL

Condition: HSML source supports the incremental framebuffer update

If the HSML Source indicates the incremental framebuffer update in the return value of the GetParameters by setting the FBUpdateOnChange bit, a SetParameters with FBUpdateOnChange bit enabled is sent to the server.

The test engineer is asked to launch an application which shows varying content.

Step	Name	Description	Expected Result
1	HSML Source Launch	See Definitions	
2	HSML Source Configuration	See Definitions bmCapabilities in SetParameters request is set to 0x00000002 (FBUpdateOnChange)	
3	Streaming Mode HSML Framebuffer Updates	Send a StartFramebufferTransmission request with the streaming mode enabled. Change the display content of the server for at least 5 seconds.	<ul style="list-style-type: none"> Valid framebuffer update events Framebuffer update event follows the framebuffer header definition in [2] Framebuffer update event is sent again automatically if there is a change in server's display content. Framebuffer content is replicating server's framebuffer
4	Intentional HSML Source Clean Up	See Definitions	

Table 20: Incremental Framebuffer Update

3.3.3 SR/HSML/BASIC/StreamingFbUpdate

Requirement: MANDATORY

Condition: None

Test if the server could send framebuffer updates continuously after receiving the StartFramebufferTransmission request with the streaming mode enabled.

The test engineer is asked to launch an application which shows varying content.

Step	Name	Description	Expected Result
1	HSML Source Launch	See Definitions	
2	HSML Source Configuration	See Definitions	
3	Streaming Mode HSML Framebuffer Updates	Send a StartFramebufferTransmission request with the streaming mode enabled.	<ul style="list-style-type: none"> Valid framebuffer update events Framebuffer update event follows the framebuffer header definition in [2]

			<ul style="list-style-type: none"> • Framebuffer update event is sent again automatically. • Framebuffer content is replicating server's framebuffer
4	Intentional HSML Source Clean Up	See Definitions	

Table 21: Streaming framebuffer transmission

3.3.4 SR/HSML/BASIC/OnDemandFbUpdate

Requirement: MANDATORY

Condition: None

Test if the server sends a framebuffer update in response to a StartFramebufferTransmission request with the on-demand mode enabled.

The test engineer is asked to launch an application which shows varying content.

Step	Name	Description	Expected Result
1	HSML Source Launch	See Definitions	
2	HSML Source Configuration	See Definitions	
3	On-demand Mode HSML Framebuffer Updates	Send a StartFramebufferTransmission request with the on-demand mode enabled. After reception of framebuffer update, send next on-demand StartFramebufferTransmission request. At least 10 iterations.	<ul style="list-style-type: none"> • A valid framebuffer update event receives • Framebuffer update event follows the framebuffer header definition in [2] • Framebuffer content is replicating server's framebuffer (visual confirmation from the test engineer)
4	Intentional HSML Source Clean Up	See Definitions	

Table 22: On-demand framebuffer transmission

3.3.5 SR/HSML/BASIC/PauseStreamingFbUpdate

Requirement: MANDATORY

Condition: None

Test if the HSML source could pause the streaming-mode framebuffer updates and resume it without reconfiguring the HSML parameters.

The test engineer is asked to launch an application which shows varying content.

Step	Name	Description	Expected Result
1	HSML Source Launch	See Definitions	
2	HSML Source Configuration	See Definitions	

3	Streaming Mode HSML Framebuffer Updates	Send a StartFramebufferTransmission request with the streaming mode enabled.	<ul style="list-style-type: none"> Valid framebuffer update events Framebuffer update event follows the framebuffer header definition in [2] Framebuffer update event is sent again automatically. Framebuffer content is replicating server's framebuffer
4	Pause the Framebuffer Updates	Send a PauseFramebufferTransmission request	<ul style="list-style-type: none"> No Framebuffer Updates receive after 1 second
5	Streaming Mode HSML Framebuffer Updates	Send a StartFramebufferTransmission request with the streaming mode enabled.	<ul style="list-style-type: none"> Valid framebuffer update events Framebuffer update event follows the framebuffer header definition in [2] Framebuffer update event is sent again automatically. Framebuffer content is replicating server's framebuffer
6	Intentional HSML Source Clean Up	See Definitions	

Table 23: Pause Streaming Framebuffer Update

3.3.6 SR/HSML/BASIC/FbHeader

Requirement: MANDATORY

Condition: None

Test if the server sends a valid framebuffer header.

Step	Name	Description	Expected Result
1	HSML Source Launch	See Definitions	
2	HSML Source Configuration	See Definitions	
3	Streaming Mode HSML Framebuffer Updates	Send a StartFramebufferTransmission request with the streaming mode enabled.	<ul style="list-style-type: none"> A valid framebuffer update events Framebuffer update event is sent again automatically. Framebuffer content is replicating server's framebuffer (visual confirmation from the test engineer) The framebuffer header MUST follow the definitions in [2]: <ul style="list-style-type: none"> Signature: 0xFFFF48534D4CFFFF Sequence number is incremental Timestamp is incremental The payload size MUST be valid

			<ul style="list-style-type: none"> Width and height are the same as the negotiated framebuffer resolution within the VNC context Pixel format and encoding follow the values in the SetParameters request. All reserved bytes are zeroes.
4	Intentional HSML Source Clean Up	See Definitions	

Table 24: Framebuffer Header

3.3.7 SR/HSML/BASIC/FrameRateAdjustment

Requirement: MANDATORY

Condition: None

Test if the server sends framebuffer updates at the rate specified in the SetMaxFrameRate request.

Step	Name	Description	Expected Result
1	HSML Source Launch	See Definitions	
2	HSML Source Configuration	See Definitions	
3	Streaming Mode HSML Framebuffer Updates	Send a StartFramebufferTransmission request with the streaming mode enabled.	<ul style="list-style-type: none"> Valid framebuffer update events Framebuffer update event follows the framebuffer header definition in [2] Framebuffer update event is sent again automatically. Framebuffer content is replicating server's framebuffer
4	Set the Maximum Frame Rate (30)	Send a SetMaxFrameRate request with value 30. Wait 1 second. Start counting the number of framebuffer updates for 3 seconds	<ul style="list-style-type: none"> The number of framebuffer updates received MUST NOT exceed 90.
5	Set the Maximum Frame Rate (20)	Send a SetMaxFrameRate request with value 20. Wait 1 second. Start counting the number of framebuffer updates for 3 seconds	The number of framebuffer updates received MUST NOT exceed 60.
6	Set the Maximum Frame Rate (10)	Send a SetMaxFrameRate request with value 10. Wait 1 second. Start counting the number of framebuffer updates for 3 seconds	The number of framebuffer updates received MUST NOT exceed 30.

7	Intentional HSML Source Clean Up	See Definitions	
---	----------------------------------	-----------------	--

Table 25: Frame rate adjustment

3.3.8 SR/HSML/BASIC/PixelFormatChange

Requirement: MANDATORY

Condition: None

Test if the server is able to change the pixel format on the fly.

Step	Name	Description	Expected Result
1	HSML Source Launch	See Definitions	
2	HSML Source Configuration	See Definitions	
3	HSML Source Start Operation	See Definitions Streaming mode is enabled	
4	Pixel Format Change (ARGB888)	Send a StopFramebufferTransmission request Send a SetParameters request with following values set: bmCapabilities: All zeroes. bPixelFormat: 0 (ARGB888) bPadding: All zeroes wPadding: All zeroes bmEncodingSupported: bit 0 (RAW) is set and all other bits are zeroes.	<ul style="list-style-type: none"> Valid framebuffer update events arrives Framebuffer update event follows the framebuffer header definition in [2] Framebuffer update event is sent again automatically. Framebuffer content is replicating server's framebuffer (visual confirmation from the test engineer) Framebuffer update using the old pixel format MUST be skipped.
5	Pixel Format Change (RGB565)	Send a StopFramebufferTransmission request Send a SetParameters request with following values set: bmCapabilities: All zeroes. bPixelFormat: 2 (RGB565) bPadding: All zeroes wPadding: All zeroes bmEncodingSupported: bit 0 (RAW) is set and all other bits are zeroes.	<ul style="list-style-type: none"> Valid framebuffer update events arrives Framebuffer update event follows the framebuffer header definition in [2] Framebuffer update event is sent again automatically. Framebuffer content is replicating server's framebuffer (visual confirmation from the test engineer) Framebuffer update using the old pixel format MAY arrives.
6	Intentional HSML Source Clean Up	See Definitions	

Table 26: Pixel Format Change

3.3.9 SR/HSML/BASIC/NonIncrementalContextInformation

Requirement: MANDATORY

- 1 Condition: None
- 2 Test if the server sends the VNC Framebuffer Update message with context information and HSML pseudo
- 3 encoding regardless of whether the context information is changed when the incremental flag within VNC
- 4 Framebuffer Update Request message is 0.

Step	Name	Description	Expected Result
1	HSML Source Launch	See Definitions	
2	HSML Source Configuration	See Definitions	
3	HSML Source Start Operation	See Definitions Streaming mode is enabled	
4	VNC Context Information	Send a non-incremental full screen VNC Framebuffer Update Request message	<ul style="list-style-type: none"> • A VNC Framebuffer Update message with two rectangles: <ul style="list-style-type: none"> ◦ Context Information ◦ HSML Pseudo encoding • Application ID is identical to UPnP value • Trust levels are valid and identical to UPnP values • Content Category is valid and identical to UPnP value • Application Category is valid and identical to UPnP value • Context information contains information for the whole framebuffer
5	Intentional HSML Source Clean Up	See Definitions	

Table 27: Non-Incremental Context Information

3.3.10 SR/HSML/BASIC/IncrementalContextInformation

Requirement: MANDATORY

Condition: None

Test if the server sends a framebuffer update messages with context information and HSML pseudo encoding when the context information is changed.

Step	Name	Description	Expected Result
1	HSML Source Launch	See Definitions	
2	HSML Source Configuration	See Definitions	
3	HSML Source Start Operation	See Definitions Streaming mode is enabled	
4	VNC Context Information	Send an incremental full framebuffer update request message Launch the next VNC based application. Send an incremental full framebuffer update request	<ul style="list-style-type: none"> • A Framebuffer Update message with two rectangles: <ul style="list-style-type: none"> ◦ Context Information ◦ HSML Pseudo encoding • Application ID is identical to UPnP value

		Continue, until all applications have been launched	<ul style="list-style-type: none"> Trust levels are valid and identical to UPnP values Content Category is valid and identical to UPnP value Application Category is valid and identical to UPnP value Context information contains information for the whole framebuffer
5	Intentional HSML Source Clean Up	See Definitions	

Table 28: Incremental Context Information

3.4 HSML Pixel Format Support

3.4.1 SR/HSML/PIXEL/ARGB888LittleEndianRAW

Requirement: MANDATORY

Condition: None

If HSML Source support the ARGB888 pixel format, a SetParameters request with ARGB888 and little endian byte order used is sent to request the server send framebuffer in desired pixel format and byte order.

Step	Name	Description	Expected Result
1	HSML Source Launch	See Definitions	
2	HSML Source Configuration	See Definitions The SetParameters is set with the following value: bmCapabilities = 0x0000 (Little endian and non-incremental) bPixelFormat = 0 (ARGB888) bmEncodingSupported = 0x00000001 (RAW).	
3	Streaming Mode HSML Framebuffer Updates	Send a StartFramebufferTransmission request with the streaming mode enabled.	<ul style="list-style-type: none"> Valid framebuffer update events Framebuffer update event follows the framebuffer header definition in [2] Framebuffer update event is sent again automatically Framebuffer content is replicating server's framebuffer (visual confirmation from the test engineer)
4	Intentional HSML Source Clean Up	See Definitions	

Table 29: ARGB 888 Pixel Format using little endian byte order - RAW

3.4.2 SR/HSML/PIXEL/ARGB888BigEndianRAW

Requirement: MANDATORY

- 1 Condition: None
- 2 If HSML Source support the ARGB888 pixel format, a SetParameters request with ARGB888 and big endian
- 3 byte order used is sent to request the server send framebuffer in desired pixel format and byte order.

Step	Name	Description	Expected Result
1	HSML Source Launch	See Definitions	
2	HSML Source Configuration	See Definitions The SetParameters is set with the following value: bmCapabilities = 0x0001 (Big endian and non-incremental) bPixelFormat = 0 (ARGB888) bmEncodingSupported = 0x00000001 (RAW).	
3	Streaming Mode HSML Framebuffer Updates	Send a StartFramebufferTransmission request with the streaming mode enabled.	<ul style="list-style-type: none"> • Valid framebuffer update events • Framebuffer update event follows the framebuffer header definition in [2] • Framebuffer update event is sent again automatically • Framebuffer content is replicating server's framebuffer (visual confirmation from the test engineer)
4	Intentional HSML Source Clean Up	See Definitions	

4 Table 30: ARGB 888 Pixel Format using big endian byte order - RAW

5 3.4.3 SR/HSML/PIXEL/RGB565LittleEndianRAW

6 Requirement: MANDATORY

7 Condition: None

- 8 If HSML Source support the RGB565 pixel format, a SetParameters request with RGB565 and little endian
- 9 byte order used is sent to request the server send framebuffer in desired pixel format and byte order.

Step	Name	Description	Expected Result
1	HSML Source Launch	See Definitions	
2	HSML Source Configuration	See Definitions The SetParameters is set with the following value: bmCapabilities = 0x0000 (Little endian and non-incremental) bPixelFormat = 2 (RGB565) bmEncodingSupported = 0x00000001 (RAW).	
3	Streaming Mode HSML Framebuffer Updates	Send a StartFramebufferTransmission request with the streaming mode enabled.	<ul style="list-style-type: none"> • Valid framebuffer update events

			<ul style="list-style-type: none"> • Framebuffer update event follows the framebuffer header definition in [2] • Framebuffer update event is sent again automatically • Framebuffer content is replicating server's framebuffer (visual confirmation from the test engineer)
4	Intentional HSML Source Clean Up	See Definitions	

Table 31: RGB 565 Pixel Format using little endian byte order - RAW

3.4.4 SR/HSML/PIXEL/RGB565BigEndianRAW

Requirement: MANDATORY

Condition: None

If HSML Source support the RGB565 pixel format, a SetParameters request with RGB565 and big endian byte order used is sent to request the server send framebuffer in desired pixel format and byte order.

Step	Name	Description	Expected Result
1	HSML Source Launch	See Definitions	
2	HSML Source Configuration	See Definitions The SetParameters is set with the following value: bmCapabilities = 0x0001 (Big endian and non-incremental) bPixelFormat = 2 (RGB565) bmEncodingSupported = 0x00000001 (RAW).	
3	Streaming Mode HSML Framebuffer Updates	Send a StartFramebufferTransmission request with the streaming mode enabled.	<ul style="list-style-type: none"> • Valid framebuffer update events • Framebuffer update event follows the framebuffer header definition in [2] • Framebuffer update event is sent again automatically • Framebuffer content is replicating server's framebuffer (visual confirmation from the test engineer)
4	Intentional HSML Source Clean Up	See Definitions	

Table 32: RGB 565 Pixel Format using big endian byte order - RAW

3.4.5 SR/HSML/PIXEL/ARGB888LittleEndianRLE

Requirement: CONDITIONAL

Condition: The HSML source supports RLE encoding

- 1 If HSML Source announced it supports RLE encoding as defined in [4], a SetParameters request with
- 2 ARGB888 and little endian byte order enabled is sent to request the server send framebuffer in desired pixel
- 3 format, byte order and encoding.

Step	Name	Description	Expected Result
1	HSML Source Launch	See Definitions	
2	HSML Source Configuration	See Definitions The SetParameters is set with the following value: bmCapabilities = 0x0000 (Little endian and non-incremental) bPixelFormat = 0 (ARGB888) bmEncodingSupported = 0x00000003 (RAW and RLE).	
3	Streaming Mode HSML Framebuffer Updates	Send a StartFramebufferTransmission request with the streaming mode enabled. Test engineer is asked to follow the steps defined in PIXIT information to let the HSML source sends RLE encoded framebuffer updates.	<ul style="list-style-type: none"> • Valid framebuffer update events • Framebuffer update event follows the framebuffer header definition in [2] • Framebuffer update event is sent again automatically • Encoding in the Framebuffer header MUST be RLE. • Pixel format in the Framebuffer header MUST be ARGB 888. • Framebuffer content is replicating server's framebuffer (visual confirmation from the test engineer)
4	Intentional HSML Source Clean Up	See Definitions	

Table 33: ARGB 888 Pixel Format using little endian byte order - RLE

3.4.6 SR/HSML/PIXEL/ARGB888BigEndianRLE

Requirement: CONDITIONAL

Condition: The HSML source supports RLE encoding

- 8 If HSML Source announced it supports RLE encoding as defined in [4], a SetParameters request with
- 9 ARGB888 and big endian byte order enabled is sent to request the server send framebuffer in desired pixel
- 10 format, byte order and encoding.

Step	Name	Description	Expected Result
1	HSML Source Launch	See Definitions	
2	HSML Source Configuration	See Definitions The SetParameters is set with the following value: bmCapabilities = 0x0001 (Big endian and non-incremental) bPixelFormat = 0 (ARGB888)	

		bmEncodingSupported = 0x00000003 (RAW and RLE).	
3	Streaming Mode HSML Framebuffer Updates	Send a StartFramebufferTransmission request with the streaming mode enabled. Test engineer is asked to follow the steps defined in PIXIT information to let the HSML source sends RLE encoded framebuffer updates.	<ul style="list-style-type: none"> Valid framebuffer update events Framebuffer update event follows the framebuffer header definition in [2] Framebuffer update event is sent again automatically Encoding in the Framebuffer header MUST be RLE. Pixel format in the Framebuffer header MUST be ARGB 888. Framebuffer content is replicating server's framebuffer (visual confirmation from the test engineer)
4	Intentional HSML Source Clean Up	See Definitions	

Table 34: ARGB 888 Pixel Format using big endian byte order - RLE

3.4.7 SR/HSML/PIXEL/RGB565LittleEndianRLE

Requirement: CONDITIONAL

Condition: The HSML source supports RLE encoding

If HSML Source announced it supports RLE encoding as defined in [4], a SetParameters request with RGB565 and little endian byte order enabled is sent to request the server send framebuffer in desired pixel format, byte order and encoding.

Step	Name	Description	Expected Result
1	HSML Source Launch	See Definitions	
2	HSML Source Configuration	See Definitions The SetParameters is set with the following value: bmCapabilities = 0x0000 (Little endian and non-incremental) bPixelFormat = 2 (RGB565) bmEncodingSupported = 0x00000003 (RAW and RLE).	
3	Streaming Mode HSML Framebuffer Updates	Send a StartFramebufferTransmission request with the streaming mode enabled. Test engineer is asked to follow the steps defined in PIXIT information to let the HSML source sends RLE encoded framebuffer updates.	<ul style="list-style-type: none"> Valid framebuffer update events Framebuffer update event follows the framebuffer header definition in [2] Framebuffer update event is sent again automatically Encoding in the Framebuffer header MUST be RLE. Pixel format in the Framebuffer header MUST be RGB 565.

			<ul style="list-style-type: none"> • Framebuffer content is replicating server's framebuffer (visual confirmation from the test engineer)
4	Intentional HSML Source Clean Up	See Definitions	

Table 35: RGB 565 Pixel Format using little endian byte order - RLE

3.4.8 SR/HSML/PIXEL/RGB565BigEndianRLE

Requirement: CONDITIONAL

Condition: The HSML source supports RLE encoding

If HSML Source announced it supports RLE encoding as defined in [4], a SetParameters request with RGB565 and big endian byte order enabled is sent to request the server send framebuffer in desired pixel format, byte order and encoding.

Step	Name	Description	Expected Result
1	HSML Source Launch	See Definitions	
2	HSML Source Configuration	See Definitions The SetParameters is set with the following value: bmCapabilities = 0x0001 (Big endian and non-incremental) bPixelFormat = 2 (RGB565) bmEncodingSupported = 0x00000003 (RAW and RLE).	
3	Streaming Mode HSML Framebuffer Updates	Send a StartFramebufferTransmission request with the streaming mode enabled. Test engineer is asked to follow the steps defined in PIXIT information to let the HSML source sends RLE encoded framebuffer updates.	<ul style="list-style-type: none"> • Valid framebuffer update events • Framebuffer update event follows the framebuffer header definition in [2] • Framebuffer update event is sent again automatically • Encoding in the Framebuffer header MUST be RLE. • Pixel format in the Framebuffer header MUST be RGB 565. • Framebuffer content is replicating server's framebuffer (visual confirmation from the test engineer)
4	Intentional HSML Source Clean Up	See Definitions	

Table 36: RGB 565 Pixel Format using big endian byte order - RLE

3.5 Blocking Notifications

3.5.1 SR/HSML/BLOCK/FbBlockingNotification

Requirement: MANDATORY

Condition: None

The server MUST follow the Framebuffer Blocking Notification mechanism defined in [4] during HSML session. The Framebuffer Blocking Notification message is still sent by the VNC client, not the HSML sink.

Step	Name	Description	Expected Result
1	HSML Source Launch	See Definitions Launch an application that is known to violate HU policies.	
2	HSML Source Configuration	See Definitions	
3	HSML Source Start Operation	See Definitions Streaming mode is enabled	
4	Context Information Update Request	Send a VNC Framebuffer Update Request message	<ul style="list-style-type: none"> • A valid VNC Framebuffer Update message receives • Only the Context Information and HSML pseudo econdings included
5	Framebuffer Blocking Notification	Send a Framebuffer Blocking Notification message with one of the following blocking reasons: <ul style="list-style-type: none"> • Not allowed content category • Not allowed application category • Not sufficient content trust level • Not sufficient application trust level • Content rules not followed • Not allowed application ID Re-Do test for each blocking reason.	<ul style="list-style-type: none"> • Phone switches its user interface to some other screen. • VNC Framebuffer update message received with new content information • Framebuffer content is replicating server's framebuffer (visual confirmation from the test engineer)
6	Intentional HSML Source Clean Up	See Definitions	

Table 37: Framebuffer Blocking Notification

4 CLIENT FEATURE TEST CASES

4.1 HSML Sink at MirrorLink Client

4.1.1 CL/HSML/USB/MultiplePersonalities

Requirement: MANDATORY

Condition: None

Test if HSML Sink recognizes both CDC/NCM and HSML at the same time.

Step	Name	Description	Expected Result
1	Physical USB cable connection.	Connect MirrorLink Client to MirrorLink Server via a USB cable	<ul style="list-style-type: none"> • MirrorLink Client is recognized as a USB host
2	MirrorLink USB command sending	Execute the steps to trigger sending the MirrorLink USB command or to manually switch to CDC/NCM and HSML support (if not done automatically)	<ul style="list-style-type: none"> • USB Device MAY receive the MirrorLink USB command • wValue is given the correct MirrorLink version (if received)
3	Switch to CDC/NCM and HSML	Provide USB CDC/NCM and HSML interface descriptors	<ul style="list-style-type: none"> • USB Host reads CDC/NCM and HSML interface descriptors • MirrorLink Client is asked to activate CDC/NCM and HSML
4	USB disconnection	Unplug the USB cable connecting Server and Client	

Table 38: Client Multiple USB Personalities Support

4.1.2 CL/HSML/USB/ShortPacket

Requirement: MANDATORY

Condition: None

Test if the HSML sink can handle USB short packets. The CTS sends Framebuffer Updates for 5 seconds. The test engineer is asked to spectate the client screen while the CTS renders a short animation. The test succeeds if the test engineer did see the animation.

Step	Name	Description	Expected Result
1	HSML Sink Launch	See Definitions	
2	Desktop Size Pseudo Encoding	Send Desktop Size Pseudo Encoding message with the following values (in separate runs of the test case): if the client framebuffer resolution within 800x480 <ul style="list-style-type: none"> • 800 x 450 • 800 x 480 if the client framebuffer resolution exceeding 800x480 <ul style="list-style-type: none"> • 854 x 480 • 1024 x 600 	

3	HSML Sink Configuration	See Definitions The wWidth and wHeight in the response to the GetParameters request are set to the value in the above Desktop Size Pseudo Encoding message	
4	HSML Sink Start Operation	See Definitions	
5	HSML Sink Framebuffer Updates	The CTS renders a short animation and send Framebuffer update continuously in the streaming mode or in response to StartFramebufferTransmission request in the on-demand mode. The test engineer is asked to spectate the client screen while a short animation is rendered.	<ul style="list-style-type: none"> The test engineer can see the animation on the client screen.
6	Intentional HSML Sink Clean Up	See Definitions	

Table 39: Client USB Short Packet

4.2 HSML Initialization Phase

4.2.1 CL/HSML/INIT/HsmIPseudoEncoding

Requirement: MANDATORY

Condition: None

Test if client sends the SetEncodings message that includes the HSML pseudo encoding.

Step	Name	Description	Expected Result
1	VNC Client Launch	See Definitions in [3].	
2	VNC Client Handshake	See Definitions in [3].	
3	VNC Client Configuration	See Definitions in [3].	<ul style="list-style-type: none"> Set Encodings includes MirrorLink pseudo encoding, Context Information pseudo encoding and HSML pseudo encoding.
4	Intentional HSML Sink Clean Up	See Definitions	

Table 40: Client HSML Pseudo Encoding

4.2.2 CL/HSML/INIT/Identification

Requirement: MANDATORY

Condition: None

- 1 Test if client sends the GetIdentifier request to retrieve the unique identifier of an HSML source, which is
- 2 generated according to the definitions in [5], after receiving the initial VNC Framebuffer Update message
- 3 that contains only the HSML pseudo encoding without any pixel data.

Step	Name	Description	Expected Result
1	HSML Sink Launch	See Definitions	
2	HSML Source Identifier Generation	Generate a UUID, as defined in [5]. Send a VNC Framebuffer Update with HSML pseudo encoding including the generated UUID.	<ul style="list-style-type: none"> • A valid GetIdentifier request receives • The wLength MUST be 16
3	Respond to GetIdentifier request	Send the generated UUID in step 2 in response to the Get Identifier request	<ul style="list-style-type: none"> • No error occurred in USB layer
4	Intentional HSML Sink Clean Up	See Definitions	

4 Table 41: Client HSML Source Identification

5 4.2.3 CL/HSML/INIT/Version

6 Requirement: MANDATORY

7 Condition: None

- 8 Test if the HSML sink sends a valid GetVersion request with the correct HSML version in wValue.

Step	Name	Description	Expected Result
1	HSML Sink Launch	See Definitions	
2	HSML Source Identifier Generation	Generate a UUID, as defined in [5]. Send a VNC Framebuffer Update with HSML pseudo encoding including the generated UUID.	<ul style="list-style-type: none"> • A valid GetIdentifier request receives • The wLength MUST be 16
3	Respond to GetIdentifier request	Send the generated UUID in step 2 in response to the Get Identifier request Wait for the GetVersion request	<ul style="list-style-type: none"> • A valid GetVersion receives • The wLength MUST be 2 • The wValue is given the correct HSML version
4	Respond to GetVersion request	Send the 0x0100	<ul style="list-style-type: none"> • No error occurred in USB layer
5	Intentional HSML Sink Clean Up	See Definitions	

9 Table 42: Client HSML Version Verification

10 4.2.4 CL/HSML/INIT/Parameters

11 Requirement: MANDATORY

12 Condition: None

1 Test if the HSML sink sends correct HSML parameters to configure the HSML source

Step	Name	Description	Expected Result
1	HSML Sink Launch	See Definitions	
2	HSML Source Identifier Generation	Generate a UUID, as defined in [5]. Send a VNC Framebuffer Update with HSML pseudo encoding including the generated UUID.	<ul style="list-style-type: none"> A valid GetIdentifier request receives The wLength MUST be 16
3	Respond to GetIdentifier request	Send the generated UUID in step 2 in response to the Get Identifier request Wait for the GetVersion request	<ul style="list-style-type: none"> A valid GetVersion receives The wLength MUST be 2 The wValue is given the correct HSML version
4	Respond to GetVersion request	Send the 0x0100 Wait for the GetParameters request	<ul style="list-style-type: none"> A valid GetParameters request receives The wLength MUST be 16
5	Respond to GetParameters request	Send the follow HSML parameters in response to GetParameters request: <ul style="list-style-type: none"> wWidth and wHeight set to 800 and 480 Pixel format ARGB888 and RGB565 Support only RAW encoding Little endian byte order used Not support the incremental update Wait for the SetParameters request	<ul style="list-style-type: none"> A valid SetParameters request receives The wLength MUST be 12 Non-incremental update is used Pixel format ARGB888 or RGB565 is selected Support RAW encoding.
6	Intentional HSML Sink Clean Up	See Definitions	

2 Table 43: Client HSML Parameters Verification

3 **4.3 HSML Basic Operation**

4 **4.3.1 CL/HSML/BASIC/NonIncrementalFbUpdate**

5 Requirement: CONDITIONAL

6 Condition: HSML sink supports streaming mode with non-incremental framebuffer updates

7 Test if the HSML sink can handle Framebuffer Updates. The CTS sends Framebuffer Updates for 5 seconds.
 8 The test engineer is asked to spectate the client screen while the CTS renders a short animation. The test
 9 succeeds if the test engineer did see the animation.

Step	Name	Description	Expected Result
1	HSML Sink Launch	See Definitions	
2	HSML Sink Configuration	See Definitions	<ul style="list-style-type: none"> FBUpdateOnChange bit is not set

3	HSML Sink Framebuffer Updates	The CTS renders a short animation and send Framebuffer update continuously in the streaming mode. The test engineer is asked to spectate the client screen while a short animation is rendered.	<ul style="list-style-type: none"> The test engineer can see the animation on the client screen.
4	Intentional HSML Sink Clean Up	See Definitions	

Table 44: Client Non-Incremental Streaming Framebuffer Update

4.3.2 CL/HSML/BASIC/IncrementalFbUpdate

Requirement: CONDITIONAL

Condition: HSML sink supports streaming mode with incremental framebuffer updates

Test if the HSML sink can handle the incremental Framebuffer Updates in the streaming mode. The CTS sends 5 Framebuffer Updates within 5 seconds by rendering a test animation that changes its content every second. The test engineer is asked to spectate the client screen while the CTS renders a short animation. The test succeeds if the test engineer did see the animation.

Step	Name	Description	Expected Result
1	HSML Sink Launch	See Definitions	
2	HSML Sink Configuration	See Definitions Support the incremental update	<ul style="list-style-type: none"> FBUpdateOnChange bit is set
3	HSML Sink Framebuffer Updates	The CTS renders the test animation that changes its content every second and send Framebuffer update continuously. The test engineer is asked to spectate the client screen while the test animation is rendered.	<ul style="list-style-type: none"> StartFramebufferTransmission message arrives Streaming Mode enabled At least 5 Framebuffer Updates received The test engineer can see the animation on the client screen.
4	Intentional HSML Sink Clean Up	See Definitions	

Table 45: Client Incremental Streaming Framebuffer Update

4.3.3 CL/HSML/BASIC/OnDemandFbUpdate

Requirement: CONDITIONAL

Condition: HSML sink supports on-demand mode

Test if the HSML sink can handle on-demand mode. The CTS sends Framebuffer Updates for 5 seconds. The test engineer is asked to spectate the client screen while the CTS renders a short animation. The test succeeds if the test engineer did see the animation.

Step	Name	Description	Expected Result
1	HSML Sink Launch	See Definitions	
2	HSML Sink Configuration	See Definitions	

3	HSML Sink Framebuffer Updates	The CTS renders a short animation and send Framebuffer updates in response to Start-FramebufferTransmission. The test engineer is asked to spectate the client screen while a short animation is rendered.	<ul style="list-style-type: none"> StartFramebufferTransmission message arrives On-demand Mode enabled The test engineer can see the animation on the client screen.
4	Intentional HSML Sink Clean Up	See Definitions	

Table 46: Client On-Demand Framebuffer Updates

4.3.4 CL/HSML/BASIC/ContextInformation

Requirement: MANDATORY

Condition: None

Test if the HSML sink responds a VNC Framebuffer Update message that includes the Context Information and HSML pseudo encodings with a valid VNC Framebuffer Update Request message to keep its context information updated.

Step	Name	Description	Expected Result
1	HSML Sink Launch	See Definitions	
2	HSML Sink Configuration	See Definitions	
3	HSML Sink Start Operation	See Definitions	
4	VNC Full Framebuffer Update Request	<ol style="list-style-type: none"> Send a VNC Framebuffer Update message including Context Information and HSML pseudo encodings Wait for a VNC Full Framebuffer Update Request message 	<ul style="list-style-type: none"> A Framebuffer Update Request message arrives The x and y are zeroes The width and height are identical to the negotiated framebuffer resolution.
5	Redo	Redo step 4 five times.	
6	Intentional HSML Sink Clean Up	See Definitions	

Table 47: Client Context Information

4.3.5 CL/HSML/BASIC/ParallelVncData

Requirement: MANDATORY

Condition: None

Test if the HSML sink correctly ignores framebuffer data being send via VNC messages, which HSML transfer is going on. The CTS will provide two different framebuffer streams, one via HSML and a second one via VNC. The test engineer will need to validate that the DUT only displays the framebuffer streamed over HSML.

Step	Name	Description	Expected Result
1	HSML Sink Launch	See Definitions	

2	HSML Sink Configuration	See Definitions	
3	HSML Sink Start Operation	See Definitions	
4	HSML Sink Framebuffer Updates	CTS renders two test images <ul style="list-style-type: none"> Test image 1 is continuously streamed over HSML. Test image 2 is continuously streamed over VNC, together with HSML and Context Information pseudo encodings. 	<ul style="list-style-type: none"> Test Image 1 is visible on the DUT screen Test image 2 is not visible on the DUT screen Framebuffer Update Request messages continuously received
5	Intentional HSML Sink Clean Up	See Definitions	

Table 48: Client receives parallel VNC Framebuffer Data

4.4 HSML Pixel Format Support

4.4.1 CL/HSML/PIXEL/ARGB888RAW

Requirement: CONDITIONAL

Condition: Client supports ARGB 888 OR

Client does not support RGB 565

Test if the HSML sink selects ARGB 888 pixel format. The test succeeds if the test engineer does see the color test image displayed on the HSML sink.

Step	Name	Description	Expected Result
1	HSML Sink Launch	See Definitions	
2	HSML Sink Configuration	See Definitions	<ul style="list-style-type: none"> Pixel format ARGB888 is selected
3	HSML Sink Start Operation	See Definitions	
4	HSML Color Test ARGB 888	Send the color test image using RAW encoding	<ul style="list-style-type: none"> The test engineer confirms the color test image displayed on the client screen.
5	Intentional HSML Sink Clean Up	See Definitions	

Table 49: ARGB 888 - RAW

4.4.2 CL/HSML/PIXEL/RGB565RAW

Requirement: CONDITIONAL

Condition: Client supports RGB 565 OR

Client does not support ARGB 888

Test if the HSML sink selects RGB 565 pixel format. The test succeeds if the test engineer does see the color test image displayed on the HSML sink.

Step	Name	Description	Expected Result
------	------	-------------	-----------------

1	HSML Sink Launch	See Definitions	
2	HSML Sink Configuration	See Definitions	<ul style="list-style-type: none"> Pixel format RGB565 is selected
3	HSML Sink Start Operation	See Definitions	
4	HSML Color Test RGB 565	Send the color test image using RAW encoding	<ul style="list-style-type: none"> The test engineer confirms the color test image displayed on the client screen.
5	Intentional HSML Sink Clean Up	See Definitions	

Table 50: RGB 565 - RAW

4.4.3 CL/HSML/PIXEL/ARGB888RLE

Requirement: CONDITIONAL

Condition: Client supports ARGB 888 OR

Client does not support RGB 565 AND

Client supports RLE encoding

Test if the HSML sink selects ARGB 888 pixel format. The test succeeds if the test engineer does see the color test image displayed on the HSML sink.

Step	Name	Description	Expected Result
1	HSML Sink Launch	See Definitions	
2	HSML Sink Configuration	See Definitions	<ul style="list-style-type: none"> Pixel format RGB565 is selected
3	HSML Sink Start Operation	See Definitions	
4	HSML Color Test ARGB 888	Send the color test image using RLE encoding	<ul style="list-style-type: none"> The test engineer confirms the color test image displayed on the client screen.
5	Intentional HSML Sink Clean Up	See Definitions	

Table 51: ARGB 888 - RLE

4.4.4 CL/HSML/PIXEL/RGB565RLE

Requirement: CONDITIONAL

Condition: Client supports RGB 565 OR

Client does not support ARGB 888 AND

Client supports RLE encoding

Test if the HSML sink selects RGB 565 pixel format. The test succeeds if the test engineer does see the color test image displayed on the HSML sink.

Step	Name	Description	Expected Result
1	HSML Sink Launch	See Definitions	

2	HSML Sink Configuration	See Definitions	<ul style="list-style-type: none"> Pixel format RGB565 is selected
3	HSML Sink Start Operation	See Definitions	
4	HSML Color Test RGB 565	Send the color test image using RLE encoding	<ul style="list-style-type: none"> The test engineer confirms the color test image displayed on the client screen.
5	Intentional HSML Sink Clean Up	See Definitions	

Table 52: RGB 565 - RLE

4.5 Blocking Notifications

4.5.1 CL/HSML/BLOCK/FbBlocked

Requirement: CONDITIONAL

Condition: Client is blocking applications

The HSML sink MUST follow the Framebuffer Blocking Notification mechanism defined in [4] during HSML session. The Framebuffer Blocking Notification message is still sent by the VNC client, not the HSML sink. This MAY be required a special procedure to trigger the client sending the Framebuffer Blocking Notification message. The test engineer MUST be provided this information prior to the test.

Step	Name	Description	Expected Result
1	HSML Sink Launch	See Definitions	
2	HSML Sink Configuration	See Definitions	
3	HSML Sink Start Operation	See Definitions	
4	Context Information Update	Wait for a VNC Framebuffer Update Request message Send a Framebuffer Update message containing allowed context information and HSML pseudo encoding	<ul style="list-style-type: none"> A valid VNC Framebuffer Update message received The x and y are zeroes The width and height are identical to the negotiated framebuffer resolution
5	Framebuffer Blocking	Send a Framebuffer Update message containing not-allowed context information and HSML pseudo encoding	<ul style="list-style-type: none"> A valid Framebuffer Blocking Notification message receives Application ID matches the context information rectangle
6	Intentional HSML Source Clean Up	See Definitions	

Table 53: Framebuffer Blocking Notification

5 REFERENCES

- [1] IETF, RFC 2119, “Keys words for use in RFCs to Indicate Requirement Levels”, March 1997.
<http://www.ietf.org/rfc/rfc2119.txt>
- [2] Car Connectivity Consortium, “MirrorLink – High Speed Media Link”, Version 1.2, CCC-TS-054
- [3] Car Connectivity Consortium, “MirrorLink – VNC based Display and Control Test Specification”, Version 1.1.3, CCC-TS-011
- [4] Car Connectivity Consortium, “MirrorLink – VNC based Display and Control”, Version 1.1, CCC-TS-10
- [5] IETF, RFC 4122, “A Universally Unique Identifier (UUID) URN Namespace”, July 2005.
<http://www.ietf.org/rfc/rfc4122.txt>