Billboard Class1.21

USB Command Verifier Compliance Test Specification

Revision 1.1

Date: December 1, 2016

Revision: 1.1

Intellectual Property Disclaimer

Revision History

Revision	Issue Date	Comments
0.90	12/18/2015	Initial release
1.1	12/1/2016	Updates to support Billboard Class v1.21

Significant Contributors:

Soren Petersen	SpecWerkz, LLC
Diane Lenox	SpecWerkz, LLC

Please send comments via electronic mail to: techadmin@usb.org or ssusbcompliance@usb.org

1 Introduction

2 Assertions to verify that USB devices are compliant with the Billboard 1.21 Specification.

General Requirements:

- All Reserved fields shall be set to 0.

Assertion #	Assertion Description	Test #
Subsection reference: 2 Management Overview		
2#1	All Device Containers that support Alternate Modes shall also support USB 2.0 at a minimum.	TD 3.1
Subsection reference: 2.1 USB Operating Speed		
2.1#1	All Device Containers that support this class shall only operate at USB 2.0 unless the Device Container already supports some form of USB functionality.	TD 3.1
2.1#2	If the Device Container includes a USB hub, then that Device Container shall expose the Billboard Device as a USB device attached to a downstream port operating at USB 2.0 speeds (real or virtual) of the hub in that Device Container.	TD 3.1
2.1#3	If the Device Container does not include a USB hub but includes some other form of USB functionality, then that Device Container shall only be required to add the descriptors and optional strings defined in Section 3 to its existing set of descriptors.	Not tested
2.1#4	If the Device Container does not include a USB hub or any other form of USB functionality, then the Device container shall be required to expose the Billboard Device as a USB device.	
2.1#5	If the Device Container includes only a billboard device with only a Billboard interface, the Device Container shall include all descriptors from Section 3.1 of Billboard Spec.	
Subsection reference: 2.2 Billboard Device Connection Process		
2.2#1	The Billboard capability shall only be exposed after Alternate Mode negotiations are completed or tAMEtimeout (as defined in Table 5.2 of [USBTYPEC]) whichever is earlier.	
2.2#2	If Port Pair does not enter into Modal Operation then the Device Container shall expose the Billboard Device.	

Assertion #	Assertion Description	Test #
2.2#3	If the Device Container uses a static list of device capabilities then it shall only expose the Billboard Device on failure to enter into Modal Operation and shall set the bmConfigured field to “Unspecified Error” (00b).	
2.2#4	If the Port Pair enters into Modal Operation but the Device Container detects a subsequent error while operating in that Mode that would cause a silent failure then the Device Container shall, if the Device Container already exposed a Billboard Device, disconnect the Billboard Device from USB, update the bmConfigured field and expose the Billboard Device.	
2.2#5	If the Port Pair enters into Modal Operation but the Device Container detects a subsequent error while operating in that Mode that would cause a silent failure then the Device Container shall, if the Device Container had not previously exposed a Billboard Device, expose the Billboard Device with the updated bmConfigured field.	
2.2#6	If the re-connection process causes disruption that is not acceptable from a user scenario point of view, the device container shall expose the Billboard Device as a standalone USB function.	
2.2#7	If a Port Pair determines that it needs to change the Alternate Mode it is operating at, the Device Container shall disconnect over USB and go back to Step 1 of section 2.2 of the USB Billboard 1.1 specification.	
2.2#8	If the Port Pair successfully enters into Modal Operation, and it wants to expose the Billboard Device, the bmConfigured field shall be set to 11b.	
2.2#9	The Billboard Device shall set the bmConfigured field to 11b upon any failure that occurs while in an Alternate Mode.	
2.2#10	Upon failure to enter the Alternate Mode within tAMEtimeout due to no USBPD communication, the Billboard Device shall set the bmConfigured field to 01b and the bAdditionalFailureInfo.bit1 to 1b.	
2.2#11	Upon failure to enter the Alternate Mode within tAMEtimeout due to insufficient power, the Billboard Device shall set the bmConfigured field to 01b and the bAdditionalFailureInfo.bit1 to 1b.	
2.2#12	Upon failure to enter the Alternate Mode within tAMEtimeout due to USBPD Alternate Mode Negotiation failure, the Billboard Device should set the bmConfigured field to 10b	

Assertion #	Assertion Description	Test #
2.2#13	If PDUSB communication as failed after the Port Pair enters into Modal Operation, the Billboard Device shall set the bmConfigured field to 11b and the bAdditionalFailureInfo.bit1 field to 1b.	
2.2#14	If a failure occurs due to lack of sufficient power while in Alternate Mode, the Billboard Device should set the bmConfigured field to 11b and the bAdditionalFailureInfo.bit0 field to 1b.	
2.2#15	If the Port Pair exits Modal Operation due to a fault condition, the Device Container shall expose the Billboard Device.	
2.2#16	Fault conditions, which cause an exit of the Alternate Mode, shall set the bmConfigured field to 01b.	
2.2#17	If USBPD communications caused exiting the Alternate Mode, the Billboard Device should set the bAdditionalFailureInfo.bit1 field to 1b.	
Subsection reference: 2.4. Support for Multiple Modes		
2.4#1	If the Device Container supports multiple modes, and Modal Configuration for all modes fails, then the Device Container is required to expose a Billboard Device.	
2.4#2	If one or more Modal Configurations succeeds, then it is optional as to whether the Device Container exposes a Billboard Device. If it does, then it shall ensure that the bmConfigured field is set appropriately for all user-facing Modes.	
Subsection reference: 3 Billboard Descriptors		
Subsection reference: 3.1 Standard Descriptors		
Subsection reference: 3.1.1 Device Descriptor		
3.1.1#1	The bcdUSB field of the Device Descriptor of a Standalone Billboard Device shall be greater than or equal to 201h.	TD 3.1
3.1.1#2	The bDeviceClass field of the Device Descriptor of a Standalone Billboard Device shall be set to 17 (11h).	Not tested
3.1.1#3	The bDeviceSubClass field of the Device Descriptor of a Standalone Billboard Device shall be set to 0.	TD 3.1
3.1.1#4	The bDeviceProtocol field of a Standalone Billboard Device shall be set to 0.	TD 3.1
3.1.1#5	It is recommended that the iManufacturer field of the Device Descriptor of a Billboard Device be set to the index of a valid string descriptor.	TD 3.1
3.1.1#6	It is recommended that the iProduct field of the Device Descriptor of a Billboard Device be set to the index of a valid string descriptor.	TD 3.1

Assertion #	Assertion Description	Test #
3.1.1#7	It is recommended that the iSerialNumber field of the Device Descriptor of a Billboard Device be set to the index of a valid string descriptor.	TD 3.1
3.1.1#8	The bNumConfigurations field of the Device Descriptor of a Standalone Billboard Device shall be set to 1.	TD 3.1
Subsection reference: 3.1.2 Device Qualifier Descriptor (if the device is a High-speed device)		
3.1.2#1	For a High Speed Device, the bcdUSB field of the Device Qualifier Descriptor of a Standalone Billboard Device shall be greater than or equal to 201h.	TD 3.1
3.1.2#2	For a High Speed Device, the bDeviceClass field of the Device Qualifier Descriptor of a Standalone Billboard Device shall be set to 17 (11h).	TD 3.1
3.1.2#3	For a High Speed Device, the bDeviceSubClass field of the Device Qualifier Descriptor of a Standalone Billboard Device shall be set to 0.	TD 3.1
3.1.2#4	For a High Speed Device, the bDeviceProtocol field of the Device Qualifier Descriptor of a Standalone Billboard Device shall be set to 0.	TD 3.1
3.1.2#5	For a High Speed Device, the bNumConfigurations field of the Device Qualifier Descriptor of a Standalone Billboard Device shall be set to 1.	TD 3.1
Subsection reference: 3.1.3 Configuration Descriptor		
3.1.3#1	The wTotalLength field of the Configuration Descriptor of a Standalone Billboard Device shall be set to 18.	TD 3.1
3.1.3#2	The bNumInterfaces field of the Configuration Descriptor of a Standalone Billboard Device shall be set to 1.	TD 3.1
3.1.3#3	It is recommended that the iConfiguration field of the Configuration Descriptor of a Billboard Device be set to the index of a valid string descriptor.	TD 3.1
Subsection reference: 3.1.4 Other_Speed_Configuration Descriptor (if the device is a High-speed device)		
3.1.4#1	For a High Speed Device, the wTotalLength field of the Other Speed Configuration Descriptor of a Standalone Billboard Device shall be set to 18.	TD 3.1
3.1.4#2	For a High Speed Device, the bNumInterfaces field of the Other Speed Configuration Descriptor of a Standalone Billboard Device shall be set to 1.	TD 3.1
3.1.4#3	For a High Speed Device, it is recommended that the iConfiguration field of the Other Speed Configuration Descriptor of a Billboard Device be set to the index of a valid string descriptor.	TD 3.1

Assertion #	**Assertion Description**	**Test #**
Subsection reference: 3.1.5 Interface Descriptor		
3.1.5#1	The bInterfaceNumber field of the Interface Descriptor of a Standalone Billboard Device shall be set to 0.	TD 3.1
3.1.5#2	The bAlternateSetting field of the Interface Descriptor of a Standalone Billboard Device shall be set to 0.	TD 3.1
3.1.5#3	The bNumEndpoints field of the Interface Descriptor of a Standalone Billboard Device shall be set to 0.	TD 3.1
3.1.5#4	The bInterfaceClass field of the Interface Descriptor of a Standalone Billboard Device shall be set to 17 (11h).	TD 3.1
3.1.5#5	The bInterfaceSubClass field of the Interface Descriptor of a Standalone Billboard Device shall be set to 0.	TD 3.1
3.1.5#6	The bInterfaceProtocol field of the Interface Descriptor of a Standalone Billboard Device shall be set to 0.	TD 3.1
3.1.5#7	It is recommended that the iInterface field of the Interface Descriptor of a Billboard Device be set to the index of a valid string descriptor.	TD 3.1
Subsection reference: 3.1.6 BOS Descriptor		
3.1.6#1	The bNumDeviceCaps field of the BOS Descriptor of a Standalone Billboard Device shall be greater than or equal to 2.	TD 3.1
Subsection reference: 3.1.6.1 Container ID		
3.1.6.1#1	A Billboard Device shall implement a Container ID capability as defined in [USB3.1] as part of the Device Container's BOS Descriptor set.	TD 3.1
Subsection reference: 3.1.6.2 Billboard Capability Descriptor		
3.1.6.2#1	A Billboard Device shall implement a Billboard Capability Descriptor as part of the Device Container's BOS Descriptor set.	TD 3.1
3.1.6.2#2	The bLength field of the Billboard Capability Descriptor shall be set to 44 + (4 * bNumberOfAlternateModes).	TD 3.1
3.1.6.2#3	The bDevCapabilityType field of the Billboard Capability Descriptor shall be set to 13 (Dh).	TD 3.1
3.1.6.2#4	It is recommended that the iAdditionalInfoURL field of the Billboard Capability Descriptor be set to the index of a valid string descriptor that contains a URL where the user can go to get more detailed information about the product and the various Alternate Modes it supports.	TD 3.1
3.1.6.2#5	The bNumberOfAlternateModes field of the Billboard Capability Descriptor shall be less than or equal to 52 (34h).	TD 3.1
3.1.6.2#6	The bPreferredAlternateMode field of the Billboard Capability Descriptor shall be strictly less than bNumberOfAlternateModes.	TD 3.1

Assertion #	Assertion Description	Test #
3.1.6.2#7	Bits 0-2 of the VCONN Power field of the Billboard Capability Descriptor shall not be set to 7 (111b).	TD 3.1
3.1.6.2#8	Bits 3-14 of the VCONN Power field of the Billboard Capability Descriptor are reserved and shall be set to 0.	TD 3.1
3.1.6.2#9	The bmConfigured field of the Billboard Capability Descriptor shall contain bNumberOfAlternateModes bit pairs.	TD 3.1
3.1.6.2#10	The bcdVersion field of the Billboard Capability Descriptor shall be set to 121h.	TD 3.1
3.1.6.2#11	Bits 2-7 of the bAdditionalFailureInfo field of the Billboard Capability Descriptor are reserved and shall be set to 0.	TD 3.1
3.1.6.2#12	The bReserved field of the Billboard Capability Descriptor is reserved and shall be set to 0.	TD 3.1
3.1.6.2#13	It is recommended that the iAlternateModeString field of an Alternate Mode contained within the Billboard Capability Descriptor contain the index of a valid String Descriptor that describes the protocol.	TD 3.1
3.1.6.2#14	Bit 0 of the bAdditionalFailureInfo Billboard Capability Descriptor shall be set to 1 if the Device Container failed due to lack of power.	
3.1.6.2#15	Bit 1 of the bAdditionalFailureInfo field of the Billboard Capability Descriptor shall be set to 1 if the Device Container failed due to no USB-PD communications.	
3.1.6.2#16	Bit 1 of the bAdditionalFailureInfo field of the Billboard Capability Descriptor is only valid if bmConfigured field is not set to 11b.	
Subsection reference: 3.1.6.3 Billboard Alternate Mode Capability Descriptor		
3.1.6.3#1	There shall be one Billboard Alternate Mode Capability Descriptor for each alternate mode supported by the Billboard Device.	TD 3.1
3.1.6.3#2	The bLength field of the Billboard Alternate Mode Capability Descriptor shall be set to 8.	TD 3.1
3.1.6.3#3	The bDescriptorType field of the Billboard Alternate Mode Capability Descriptor shall be set to 0fh	
3.1.6.3#4	The bIndex field of the Billboard Alternate Mode Capability Descriptor shall be the index of where the Alternate Mode appears in the array of Alternate Modes described in the Billboard Capability Descriptor.	TD 3.1
3.1.6.3#5	The dwAlternatemodeVdo field of the Billboard Alternate Mode Capability Descriptor shall be the contents of the Mode VDO for the alternate mode identified by the bIndex field.	

3. Test Descriptions for Billboard

General Test Initialization

All Billboard Device tests follow the same initialization procedure. At the beginning of a test run, the host controller is reset and devices attached to the host are enumerated. If attached devices fail initialization assertions, the test tool will not be able to run the test. Note that the host controller is not reset in between tests if more than 1 test is selected.

Assertions Used in Test Initialization

Chapter 9 Assertions: 9.1.1#1, 9.1.1#2, 9.1.1.3#1, 9.1.1.4#1, 9.1.1.5#1, 9.2.6.1#1, 9.2.6.3#1 9.2.6.4#1, 9.3#1, 9.4.3#5, 9.4.3#6 9.4.3#10, 9.4.3#10, 9.4.3#11

TD 3.1 Billboard Descriptors Test

Assertions Used in Test

2#1, 2.1#1, 2.1#2, 3.1.1#1—3.1.1#8, 3.1.2#1—3.1.2#5, 3.1.3#1—3.1.3#3, 3.1.4#1—3.1.4#3, 3.1.5#1—3.1.5#7, 3.1.6#1, 3.1.6.1#1, 3.1.6.2#1—3.1.6.2#13, 3.1.6.3#1, 3.1.6.3#2, 3.1.6.3#4

Device States for Test

This test is run with the device in the configured state.

Overview of Test Steps

The test software performs the following steps.

1. Retrieve the Device Descriptor of the Device Under Test by sending a GetDescriptor(Device) request.
    a. Test aborts if request fails.
    b. Test fails if *bcdUSB* field is less then 0x0200 (2#1)
    c. Test fails if *bcdUSB* field is less then 0x0201 (3.1.1#1)
    d. Issue warning if *iManufacturer* field is zero or GetDescriptor(String, *iManufacturer*) fails. (3.1.1#5)
    e. Issue warning if *iProduct* field is zero or GetDescriptor(String, *iProduct*) fails. (3.1.1#6)
    f. Issue warning if *iDeviceClass* field of the Device Descriptor.
    g. If *bDeviceClass* equals 17, then the ***Standalone*** Billboard device.
2. If Device Under Test is not a Standalone Billboard device, examine the Device Descriptor:
    a. Test fails if Device Under Test is a Hub (*bDeviceType* = 9) (2.1#2)
3. If Device Under Test is a Standalone Billboard device, examine Device Descriptor:
    a. Test fails if device speed is SuperSpeed or above. (2.1#1)
    b. Test fails if *bDeviceSubClass* is not equal to 0. (3.1.1#3)
    c. Test fails if *bDeviceProtocol* is not equal to 0. (3.1.1#4)
    d. Test fails if *bNumConfigerations* is not equal to 1. (3.1.1#8)
4. If Device Under Test is a High Speed Standalone Billboard device, retrieve the Device Qualifier Descriptor of Device Under Test by sending a GetDescriptor(DeviceQualifier) request.
    a. Test records an abort if request fails, but continues with step 5.
    b. Test fails if *bcdUSB* is less than 0x201. (3.1.2#1)
    c. Test fails if *bDeviceClass* is not equal to 17. (3.1.2#2)
    d. Test fails if *bDeviceSubClass* of is not equal to 0. (3.1.2#3)
    e. Test fails if *bDeviceProtocol* is not equal to 0. (3.1.2#4)
    f. Test fails if *bNumConfigerations* field is not equal to 1. (3.1.2#5)
5. Retrieve the full Configuration Descriptor of the Device Under Test by first sending a GetDescriptor(Configuration) with a length of 9, and then sending a GetDescriptor(Configuration) with a length of the *wTotalLength* field of the Configuration Descriptor.
    a. Test records an abort if either request fails, but continues with step 11.
    b. Issue warning if *iConfiguration* field is zero, or GetDescriptor(String, *iConfiguration*) fails. (3.1.3#3)
6. If Device Under Test is a Standalone Billboard device, examine Configuration Descriptor.
    a. Test fails if *wTotalLength* is not equal to 18. (3.1.3#1)
    b. Test fails if *bNumInterfaces* is not equal to 1. (3.1.3#2)
7. Parse the full Configuration Descriptor, retrieving all Interface Descriptors. Examine each Descriptor:

    a. Issue warning if *iInterface* field is zero, or GetDescriptor(String, *iInterface*) fails. (3.1.5#7)

8. If Device Under Test is a Standalone Billboard device, examine each Interface Descriptor:
    a. Test fails if *bInterfaceNumber* is not equal to 0. (3.1.5#1)
    b. Test fails if *bAlternateSetting* is not equal to 0. (3.1.5#2)
    c. Test fails if *bNumEndpoints* is not equal to 0. (3.1.5#3)
    d. Test fails if *bInterfaceClass* is not equal to 17. (3.1.5#4)
    e. Test fails if *bInterfaceSubClass* of is not equal to 0. (3.1.5#5)
    f. Test fails if *bInterfaceProtocol* is not equal to 0. (3.1.5#6)
9. If Device Under Test is a High Speed Device, retrieve the full Other Speed Configuration Descriptor of the Device Under Test by first sending a GetDescriptor(Other_Speed_Configuration) with a length of 9, and then sending a GetDescriptor(Other_Speed_Configuration) with a length of the *wTotalLength* field of the Other Speed Configuration Descriptor.
    a. Test records an abort if either request fails, but continues with step 13.
    b. Issue warning if *iConfiguration* field is zero, or GetDescriptor(String, *iConfiguration*) fails. (3.1.4#3)
10. If Device Under Test is a High Speed Standalone Billboard device, examine Other Speed Configuration Descriptor.
    a. Test fails if *wTotalLength* is not equal to 18. (3.1.4#1)
    b. Test fails if *bNumInterfaces* is not equal to 1. (3.1.4#2)
11. If Device Under Test is a High Speed Standalone Billboard device, parse the full Other Speed Configuration Descriptor, retrieving all Interface Descriptors. Examine each Descriptor:
    a. Issue warning if *iInterface* field is zero, or GetDescriptor(String, *iInterface*) fails. (3.1.5#7)
12. If Device Under Test is a High Speed Standalone Billboard device, examine each Interface Descriptor in the full Other Speed Configuration Descriptor:
    a. Test fails if *bInterfaceNumber* is not equal to 0. (3.1.5#1)
    b. Test fails if *bAlternateSetting* is not equal to 0. (3.1.5#2)
    c. Test fails if *bNumEndpoints* is not equal to 0. (3.1.5#3)
    d. Test fails if *bInterfaceClass* is not equal to 17. (3.1.5#4)
    e. Test fails if *bInterfaceSubClass* of is not equal to 0. (3.1.5#5)
    f. Test fails if *bInterfaceProtocol* is not equal to 0. (3.1.5#6)
13. Retrieve the full BOS Descriptor of the Device Under Test by first sending a GetDescriptor(BOS) with a length of 9, and then sending a GetDescriptor(BOS) with a length of the *wTotalLength* field of the BOS Descriptor.
    a. Test aborts if either request fails.
    b. Test fails if *bNumDeviceCaps* is not greater than or equal to 2. (3.1.6#1)
14. Parse the full BOS Descriptor, retrieving the Container ID Capability Descriptor.
    a. Test fails if there is no Container ID Capability Descriptor. (3.1.6.1#1)
15. Parse the full BOS Descriptor, retrieving the Billboard Capability Descriptor.
    a. Test fails if there is no Billboard Capability Descriptor (3.1.6.2#1)
    b. Test fails if *bLength* is not equal to 44 + (4 * *bNumberOfAlternateModes*). (3.1.6.2#2)
    c. Test fails if *bDevCapabilityType* is not equal to 13. (3.1.6.2#3)
    d. Issue a warning if *iAdditionalInfoURL* equals 0, or GetDescriptor(String, *iAdditionalInfoURL*) fails. (3.1.6.2#4)
    e. Test fails if *bNumberOfAlternateModes* is greater than 52. (3.1.6.2#5)
    f. Test fails if *bPreferredAlternateMode* is greater than or equal to *bNumberOfAlternateModes*. (3.1.6.2#6)
    g. Test fails if bits 0-2 of the VCONN power field are equal to 7. (3.1.6.2#7)
    h. Test fails if bits 3-14 of the VCONN power field are not equal to 0. (3.1.6.2#8)
    i. Test fails if *bmConfigured* contains more than *bNumberOfAlternateModes* bit pairs. (3.1.6.2#9)
    j. Test fails if bcdVersion is not equal to 121h. (3.1.6.2#10)
    k. Test fails if bits 2-7 of *bAdditionalFailureInfo* are not equal to 0. (3.1.6.2#11)
    l. Test fails if *bReserved* is not equal to 0.
    m. For each Alternate Mode contained in the Billboard Capability Descriptor.
        i. Issue a warning if *iAlternateModeString* equals 0, or GetDescriptor(String, *iAlternateModeString*) fails. (3.1.6.2#12)
16. Parse the full BOS Descriptor, retrieving all Billboard Alternate Mode Capability Descriptors.

a. Test fails if the number of Billboard Alternate Mode Capability Descriptors does not match bNumberOfAlternateModes field in Billboard Capability descriptor (3.1.6.3#1)
b. For each Billboard Alternate Mode Capability Descriptor:
    i. Test fails if bLength is not equal to 8 (3.1.6.3#2).
    ii. Test fails if the bIndex field is greater than or equal to bNumberOfAlternateModes field of Billboard Capability Descriptor. (3.1.6.3#4).
    iii. Test fails if there is more than 1 Billboard Alternate Mode Capability Descriptor with the same bIndex value (3.1.6.3#1).

Appendix

Changes in Revision 0.90

1. Initial Check-in.

Changes in Revision 1.1

1. Updates for revision 1.21 of Billboard Class Specification.