

Universal Serial Bus Device Class Definition for Billboard Devices

**Revision 1.21
September 8, 2016**

**Copyright © 2016, USB 3.0 Promoter Group
All rights reserved.**

INTELLECTUAL PROPERTY DISCLAIMER

THIS SPECIFICATION IS PROVIDED TO YOU "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE. THE AUTHORS OF THIS SPECIFICATION DISCLAIM ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF ANY PROPRIETARY RIGHTS, RELATING TO USE OR IMPLEMENTATION OF INFORMATION IN THIS SPECIFICATION. THE PROVISION OF THIS SPECIFICATION TO YOU DOES NOT PROVIDE YOU WITH ANY LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS.

All product names are trademarks, registered trademarks, or service marks of their respective owners.

CONTENTS

Specification Work Group Contributors	4
1 Introduction	5
1.1 Scope.....	5
1.2 Purpose	5
1.3 Related Documents.....	5
1.4 Terms and Abbreviations	5
1.5 Conventions and Notations.....	5
1.5.1 Precedence	5
1.5.2 Keywords	5
1.5.3 Numbering.....	6
1.5.4 Byte Ordering	6
2 Management Overview	7
2.1 USB Operating Speed	7
2.2 Billboard Device Connection Process	7
2.3 Non user-facing Modes.....	9
2.4 Support for Multiple Modes.....	9
2.5 Use of URLs	9
3 Billboard Descriptors	10
3.1 Standard Descriptors.....	10
3.1.1 Device Descriptor	10
3.1.2 Device Qualifier Descriptor (if the device is a High-speed device) 10	
3.1.3 Configuration Descriptor	11
3.1.4 Other_Speed_Configuration Descriptor (if the device is a High-speed device)	11
3.1.5 Interface Descriptor	11
3.1.6 BOS Descriptor	12
4 Requests and Control Sequences	17
4.1 Standard Requests.....	17
4.2 Class-Specific Requests.....	17
A Values of Constants.....	18
A.1. Billboard Class Code	18
A.2. Billboard Subclass Codes.....	18
A.3. Billboard Protocol Codes.....	18
A.4. Billboard Descriptor Types.....	18
A.5. Parameter Values.....	18

TABLES

Table 1-1: Terms and Abbreviations	5
Table 3-1: Device Descriptor.....	10
Table 3-2: Device Qualifier Descriptor	10
Table 3-3: Configuration Descriptor	11
Table 3-4: Other_Speed_Configuration Descriptor.....	11

Table 3-5: Interface Descriptor	11
Table 3-6: BOS Descriptor	12
Table 3-7: Billboard Capability Descriptor	12
Table 3-8 Example Values for wSVID, bAlternateMode, iAlternateModeString.....	14
Table 3-9: Billboard Alternate Mode Capability Descriptor.....	15
Table 3-10 Example Billboard Alternate Mode Capability Descriptors	15
Table 4-1 Billboard Class Code.....	18
Table 4-2 Billboard Subclass Code	18
Table 4-3 Billboard Protocol Code.....	18
Table 4-4 Billboard Capability Descriptor Types.....	18
Table 4-5 Parameter Values	18

Specification Work Group Contributors

Advanced Micro Devices	Jim Hunkins		
Apple	Colin Whitby-Strevens	Dan Wilson	Reese Schreiber
	Scott Deandrea	Scott Jackson	
DisplayLink (UK) Ltd.	Richard Petrie		
Google	David Schneider	Mark Hayter	Vincent Palatin
Intel Corporation	Abdul (Rahman) Ismail – Editor	Bob Dunstan	Brad Saunders
	Christine Krause	David Hines	Rafal Wielicki
Microsoft Corporation	Jayson Kastens	Randy Aull	Vivek Gupta
NXP Semiconductors	Bart Vertenten		

1 Introduction

1.1 Scope

The USB Billboard Device Class definition describes the methods used to communicate the Alternate Modes supported by a Device Container to a host system. This includes string descriptors that can be used to provide support details in a human-readable format.

This specification does not describe the functionality/methodology by which the Device Container shall switch to an Alternate Mode. An example of a Device Container that shall support this class is one that supports PCIe over the USB Type-C connector.

1.2 Purpose

The purpose of this document is to describe the minimum capabilities and characteristics a Billboard Device shall support to be compliant.

1.3 Related Documents

- [USB2.0] – Universal Serial Bus Specification, Revision 2.0, (including errata and ECNs through August 11, 2014) (referred to in this document as the USB 2.0 Specification) (available at: <http://www.usb.org/developers/docs>).
- [USB3.1] – Universal Serial Bus 3.1 Specification, Revision 1.0, (including errata and ECNs through March 1, 2016) (referred to in this document as the USB 3.1 Specification) (available at: <http://www.usb.org/developers/docs>.)
- [USBPD] – Universal Serial Bus Power Delivery Specification, Revision 2.0, March 25, 2016 (referred to in this document as the USB PD Specification) (available at: <http://www.usb.org/developers/docs>.)
- [USBTYPEC] – Universal Serial Bus Universal Serial Bus Type-C Cable and Connector Specification, Revision 1.2, March 25, 2016 (referred to in this document as the USB Type-C Specification) (available at: <http://www.usb.org/developers/docs>.)

1.4 Terms and Abbreviations

This section defines terms and abbreviations used throughout this document. For additional terms and abbreviations that pertain to the Universal Serial Bus, see Chapter 2, “Terms and Abbreviations,” in [USB2.0] and [USB3.1], Section 1.5 in [USBTYPEC] and Section 1.6 in [USBPD].

Table 1-1: Terms and Abbreviations

Term	Description
Billboard Device	Either a standalone USB device that adheres to this class specification or a Device Container that exposes other USB functionality but includes the Billboard Descriptors as part of its complete BOS descriptor
Device Container	A group of one or more USB functions originated from the same physical device

1.5 Conventions and Notations

1.5.1 Precedence

If there is a conflict between text, figures, and tables, the precedence shall be tables, figures, and then text.

1.5.2 Keywords

The following keywords differentiate between the levels of requirements and options.

1.5.2.1 Informative

Informative is a keyword that describes information with this specification that intends to discuss and clarify requirements and features as opposed to mandating them.

1.5.2.2 May

May is a keyword that indicates a choice with no implied preference.

1.5.2.3 N/A

N/A is a keyword that indicates that a field or value is not applicable and has no defined value and shall not be checked or used by the recipient.

1.5.2.4 Normative

Normative is a keyword that describes features that are mandated by this specification.

1.5.2.5 Optional

Optional is a keyword that describes features not mandated by this specification. However, if an optional feature is implemented, the feature shall be implemented as defined by this specification (optional normative).

1.5.2.6 Reserved

Reserved is a keyword indicating reserved bits, bytes, words, fields, and code values that are set-aside for future standardization. Their use and interpretation may be specified by future extensions to this specification and, unless otherwise stated, shall not be utilized or adapted by vendor implementation. A reserved bit, byte, word, or field shall be set to zero by the sender and shall be ignored by the receiver. Reserved field values shall not be sent by the sender and, if received, shall be ignored by the receiver.

1.5.2.7 Shall

Shall is a keyword indicating a mandatory (normative) requirement. Designers are mandated to implement all such requirements to ensure interoperability with other compliant Devices.

1.5.2.8 Should

Should is a keyword indicating flexibility of choice with a preferred alternative. Equivalent to the phrase "it is recommended that".

1.5.3 Numbering

Numbers that are immediately followed by a lowercase "b" (e.g., 01b) are binary values. Numbers that are immediately followed by an uppercase "B" are byte values. Numbers that are immediately followed by a lowercase "h" (e.g., 3Ah) are hexadecimal values. Numbers not immediately followed by either a "b", "B", or "h" are decimal values.

1.5.4 Byte Ordering

All multiple byte fields in this specification are interpreted as and moved over the bus in little-endian order, i.e., LSB to MSB unless otherwise specified.

2 Management Overview

The USB framework is well suited for describing a product and the intent of this specification is to describe what Alternate modes that this Device Container supports. All Device Containers that support Alternate Modes shall also support USB 2.0 at a minimum.

USB has become a ubiquitous connector on modern PC's and mobile devices and is well understood by most consumers today. This class ensures that Device Containers that support Alternate Modes that are connected to a host that doesn't support one or more of the Alternate Modes has an easy route by which the host can report the same to the user without any silent failures.

It is recommended to return strings that describe the Device Container and the Alternate Modes it supports, however, it is optional and the choice of languages is left up to the Device Container implementer.

2.1 USB Operating Speed

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.

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.

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 3.1.6 to its existing set of descriptors.

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.

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.

2.2 Billboard Device Connection Process

1. The Billboard capability shall only be exposed after Alternate Mode negotiations are completed or tAMetimeout (as defined in Table 5.2 of [USBTYPPEC]) whichever is earlier.
2. If the Port Pair successfully enters into Modal Operation (as defined by [USBPD]) then the Device Container is not required to expose the Billboard Device. It may still expose the Billboard Device if it wants to for informational purposes only.
 - a. 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.
 - b. The Billboard Device shall set the **bmConfigured** field to 11b upon any failure that occurs while in an Alternate Mode.
3. If the Port Pair does not enter into Modal Operation, then the Device Container shall expose the Billboard Device.
 - a. 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.

- b. Upon failure to enter the Alternate Mode within `tAMTimeout` due to insufficient power, the Billboard Device shall set the **bmConfigured** field to 01b and the **bAdditionalFailureInfo.bit0** to 1b.
 - c. Upon failure to enter the Alternate Mode within `tAMTimeout` due to [USBPD] Alternate Mode negotiation failure, the Billboard Device should set the **bmConfigured** field to 10b.
 4. 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).
 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:
 - a. If the Device Container already exposed a Billboard Device then it shall disconnect the Billboard Device from USB, update the **bmConfigured** field and expose the Billboard Device.
 - b. If the Device Container had not previously exposed a Billboard Device, then it shall expose the Billboard Device with the updated **bmConfigured** field.
 - c. If [USBPD] communication has failed, the Billboard Device shall set the **bmConfigured** field to 11b and the **bAdditionalFailureInfo.bit1** field to 1b.
 - d. If the failure has occurred due to lack of sufficient power while in the Alternate Mode, the Billboard Device should set the **bmConfigured** field to 11b and the **bAdditionalFailureInfo.bit0** to 1b. This failure can occur at any time while in an Alternate Mode. Possible scenarios where this might occur are as follows: the AMA requires more power and attempts to negotiate this via [USBPD] but this fails, the power contract has been negotiated via [USBPD] with mismatch and the AMA can't fully function, the power cord being unplugged from the wall socket.
 6. If the Port Pair exits Modal Operation due to a fault condition, the Device Container shall expose the Billboard Device
 - a. If the Device Container already exposed a Billboard Device then it shall disconnect the Billboard Device from USB, update the **bmConfigured** field and expose the Billboard Device.
 - b. If the Device Container had not previously exposed a Billboard Device, then it shall expose the Billboard Device with the updated **bmConfigured** field.
 - c. Fault conditions, which cause an exit of the Alternate Mode, shall set the **bmConfigured** field to 01b.
 - d. If [USBPD] communication caused exiting the Alternate Mode, the Billboard Device should set the **bAdditionalFailureInfo.bit1** field to 1b.

If the device supports USB functionality, this re-connection process can cause disruption with the already exposed USB function. If that disruption is not acceptable from user scenario point of view, the device container shall expose the Billboard Device as a standalone USB function.

1. 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.

Note that it may take additional time to negotiate a USB PD Explicit Contract between a Port Pair and hence this may lead to a significant delay in the enumeration of the Billboard Device.

2.3 Non user-facing Modes

A Device Container may support non user-facing Modes (e.g. diagnostic modes). When Modal Operation fails for a non-user-facing Mode, the Device Container should not expose a Billboard Device, and, if it does, it should not report the failure in the **bmConfigured** field.

2.4 Support for Multiple Modes

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. 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.

2.5 Use of URLs

The URL given by **iAdditionalInfoURL** in the Billboard Capability Descriptor (offset 3) is intended to be the URL of a web page providing descriptive useful information concerning the device and all the user-facing Alternate Modes that it supports. This information is not intended specifically to provide help when an Alternate Mode Configuration fails.

iAlternateModeString[n] for a specific Alternate Mode for a specific SVID is intended, as well as describing the protocol, to contain a URL of a web page providing “help” information to be displayed by the Host when entry to the Alternate Mode fails. An example string is DisplayPort to VGA adapter. For further assistance, see <http://help.vesa.org/dp-usb-type-c/>.

3 Billboard Descriptors

The Billboard Descriptors are used to advertise the Alternate Modes a Device Container supports through a BOS descriptor capability.

3.1 Standard Descriptors

A standalone Billboard Device is always defined at the device level and does not support any other endpoints apart from the default control endpoint. Therefore, it only supports the standard Device, Device Qualifier (only if it is a High-speed device), BOS, Configuration and Other_Speed_Configuration (only if it is a High-speed device) descriptors. And it uses standard USB mechanisms to return the descriptors.

The subsequent sections define the Device, Device Qualifier and Configuration descriptors a standalone Billboard Device shall return.

Note, a Device Container that supports Alternate Modes and one or more other USB Device functions (apart from a Device Container that exposes a USB hub) only needs to add the Container ID and Billboard Descriptor capability to the list of capabilities it returns when queried for its BOS descriptor.

For a detailed description of the various fields in the tables in the following sections please refer to the definitions in [USB2.0] and [USB3.1].

3.1.1 Device Descriptor

Table 3-1: Device Descriptor

Offset	Field	Bytes	Value	Description
0	bLength	1	Number	Size of this Descriptor in bytes: 18.
1	bDescriptorType	1	Constant	DEVICE_DESCRIPTOR Type.
2	bcdUSB	2	BCD	0x201H (minimum)
4	bDeviceClass	1	Class	BILLBOARD_CLASS. See Appendix
5	bDeviceSubClass	1	SubClass	BILLBOARD_SUBCLASS. See Appendix
6	bDeviceProtocol	1	Protocol	BILLBOARD_PROTOCOL. See Appendix
7	bMaxPacketSize0	1	Number	Maximum packet size for endpoint zero. (Only 8, 16, 32, or 64 are valid)
8	idVendor	2	ID	Vendor ID
10	idProduct	2	ID	Product ID
12	bcdDevice	2	BCD	Device release number in binary-coded decimal
14	iManufacturer	1	Index	Index of string descriptor describing manufacturer
15	iProduct	1	Index	Index of string descriptor describing product
16	iSerialNumber	1	Index	Index of string descriptor describing the device's serial number
17	bNumConfigurations	1	Number	1

3.1.2 Device Qualifier Descriptor (if the device is a High-speed device)

Table 3-2: Device Qualifier Descriptor

Offset	Field	Bytes	Value	Description
0	bLength	1	Number	Size of this Descriptor in bytes: 10.
1	bDescriptorType	1	Constant	DEVICE_QUALIFIER_DESCRIPTOR Type.
2	bcdUSB	2	BCD	0x201H (minimum)
4	bDeviceClass	1	Class	BILLBOARD_CLASS. See Appendix
5	bDeviceSubClass	1	SubClass	BILLBOARD_SUBCLASS. See Appendix
6	bDeviceProtocol	1	Protocol	BILLBOARD_PROTOCOL. See Appendix

Offset	Field	Bytes	Value	Description
7	bMaxPacketSize0	1	Number	Maximum packet size for endpoint zero for other speed.
8	bNumConfigurations	1	Number	1 (Number of Other-speed Configurations)
9	bReserved	1	Zero	Reserved for future use, shall be set to zero

3.1.3 Configuration Descriptor

Table 3-3: Configuration Descriptor

Offset	Field	Bytes	Value	Description
0	bLength	1	Number	Size of this Descriptor in bytes: 9.
1	bDescriptorType	1	Constant	CONFIGURATION Descriptor Type
2	wTotalLength	2	Number	Total length of data returned for this configuration.
4	bNumInterfaces	1	Number	1.
5	bConfigurationValue	1	Number	Values to use to select this configuration
6	iConfiguration	1	Index	Index of string descriptor describing this configuration
7	bmAttributes	1	Bitmap	Shall report whether the Device Container is Self-powered or not.
8	bMaxPower	1	mA	Maximum power consumption of this Device Container from VBUS.

3.1.4 Other_Speed_Configuration Descriptor (if the device is a High-speed device)

Table 3-4: Other_Speed_Configuration Descriptor

Offset	Field	Bytes	Value	Description
0	bLength	1	Number	Size of this Descriptor in bytes: 9.
1	bDescriptorType	1	Constant	OTHER_SPEED_CONFIGURATION Descriptor Type
2	wTotalLength	2	Number	Total length of data returned.
4	bNumInterfaces	1	Number	1.
5	bConfigurationValue	1	Number	Values to use to select this configuration
6	iConfiguration	1	Index	Index of string descriptor describing this configuration
7	bmAttributes	1	Bitmap	Same as Configuration descriptor.
8	bMaxPower	1	mA	Same as Configuration descriptor.

3.1.5 Interface Descriptor

Table 3-5: Interface Descriptor

Offset	Field	Bytes	Value	Description
0	bLength	1	Number	Size of this descriptor in bytes: 9
1	bDescriptorType	1	Constant	INTERFACE Descriptor Type
2	bInterfaceNumber	1	Number	0
3	bAlternateSetting	1	Number	0
4	bNumEndpoints	1	Number	0
5	bInterfaceClass	1	Class	BILLBOARD_CLASS. See Appendix
6	bInterfaceSubClass	1	SubClass	BILLBOARD_SUBCLASS. See Appendix
7	bInterfaceProtocol	1	Protocol	BILLBOARD_PROTOCOL. See Appendix
8	iInterface	1	Index	Index of string descriptor describing this interface

3.1.6 BOS Descriptor

Table 3-6: BOS Descriptor

Offset	Field	Bytes	Value	Description
0	bLength	1	Number	Size of descriptor:5
1	bDescriptorType	1	Constant	BOS Descriptor type
2	wTotalLength	2	Number	Length of this descriptor and all of its sub descriptors
4	bNumDeviceCaps	1	Number	The number of separate device capability descriptors in the BOS At a minimum, the device shall have to return the Billboard Capability and Container ID descriptors as part of its BOS descriptor.

3.1.6.1 Container ID

A Billboard Device shall implement a Container ID capability as defined in [USB3.1]. This capability must be returned by Billboard Devices operating at USB 2.0 speeds as well. The value in the **ContainerID** field may be used by software to uniquely identify the components of the Device Container when it is connected to the system.

3.1.6.2 Billboard Capability Descriptor

This descriptor lists the Alternate Modes the Device Container supports and optional string for each Alternate Mode supported. This descriptor is returned as part of the Device Container's BOS Descriptor set.

Table 3-7: Billboard Capability Descriptor

Offset	Field	Size	Value	Description
0	bLength	1	Number	Size of this Descriptor in bytes.
1	bDescriptorType	1	Constant	Descriptor type: DEVICE CAPABILITY Type.
2	bDevCapabilityType	1	Number	BILLBOARD CAPABILITY
3	iAdditionalInfoURL	1	Index	Index of string descriptor providing a URL where the user can go to get more detailed information about the product and the various Alternate Modes it supports.
4	bNumberOfAlternateModes	1	Number	Number of Alternate modes supported. The maximum value that this field can be set to is MAX_NUM_ALT_MODE.
5	bPreferredAlternateMode	1	Number	Index of the preferred Alternate Mode. System software may use this information to provide the user with a better user experience.

Offset	Field	Size	Value	Description										
6	VCONN Power	2	Bitmap	<table border="1"> <thead> <tr> <th>Bit</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>2..0</td> <td>V_{CONN} Power needed by the adapter for full functionality 000b = 1W 001b = 1.5W 010b = 2W 011b = 3W 100b = 4W 101b = 5W 110b = 6W 111b = reserved</td> </tr> <tr> <td>14..3</td> <td>Reserved for future use, shall be set to zero.</td> </tr> <tr> <td>15</td> <td>The adapter does not require any V_{CONN} Power. The value in Bits 2..0 are ignored when this bit is set.</td> </tr> </tbody> </table>	Bit	Description	2..0	V _{CONN} Power needed by the adapter for full functionality 000b = 1W 001b = 1.5W 010b = 2W 011b = 3W 100b = 4W 101b = 5W 110b = 6W 111b = reserved	14..3	Reserved for future use, shall be set to zero.	15	The adapter does not require any V _{CONN} Power. The value in Bits 2..0 are ignored when this bit is set.		
Bit	Description													
2..0	V _{CONN} Power needed by the adapter for full functionality 000b = 1W 001b = 1.5W 010b = 2W 011b = 3W 100b = 4W 101b = 5W 110b = 6W 111b = reserved													
14..3	Reserved for future use, shall be set to zero.													
15	The adapter does not require any V _{CONN} Power. The value in Bits 2..0 are ignored when this bit is set.													
8	bmConfigured	32	Bitmap	<p>Each bit pair in this field indicates the state of the Alternate Modes identified by the combination of wSVID[i] and bAlternateMode[i]. A Device Container may support a maximum of MAX_NUM_ALT_MODE Alternate Modes. System software will determine what to display, if anything, based on the value of the bits in this field.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Unspecified Error</td> </tr> <tr> <td>01b</td> <td>Alternate Mode configuration not attempted or exited</td> </tr> <tr> <td>10b</td> <td>Alternate Mode configuration attempted but unsuccessful and not entered</td> </tr> <tr> <td>11b</td> <td>Alternate Mode configuration successful</td> </tr> </tbody> </table> <p>This bitmap corresponds to the Alternate Mode as follows: Bit 0: First Alternate Mode defined (at Offset 44) Bit 2: Second Alternate Mode defined (at Offset 48) Bit (n-1) * 2: Nth Alternate Mode defined</p>	Value	Description	00b	Unspecified Error	01b	Alternate Mode configuration not attempted or exited	10b	Alternate Mode configuration attempted but unsuccessful and not entered	11b	Alternate Mode configuration successful
Value	Description													
00b	Unspecified Error													
01b	Alternate Mode configuration not attempted or exited													
10b	Alternate Mode configuration attempted but unsuccessful and not entered													
11b	Alternate Mode configuration successful													
40	bcdVersion	2	BCD	<p>Billboard Capability version number in Binary-Coded Decimal (e.g., 1.10 is 0x0110H). This field identifies the release of the Billboard Specification with which the Billboard descriptors are compliant. All Device Containers compliant to this version of the specification shall set the value to 0x0121H. A value of 0x0000H indicates that the device was built to the first version of this specification. A value of 0x0120H in this field is reserved.</p>										

Offset	Field	Size	Value	Description	
				Bit	Description
42	bAdditionalFailureInfo	1	Bitmap	0	If this field is set to one then the Device Container failed due to lack of power.
				1	If this field is set to one then the Device Container failed due to no USB-PD communication. This field is only valid if bmConfigured field for the preferred Alternate mode is not set to 11b.
				7..2	Reserved for future use, shall be set to zero.
				This field is only valid if the bcdVersion field is set to 0x0110h or higher.	
43	bReserved	1	Zero	Reserved for future use, shall be set to zero	
44	wSVID[0]	2	Number	Standard or Vendor ID. This shall match one of the SVIDs returned in response to a USB PD Discover SVIDs command.	
46	bAlternateMode[0]	1	Number	Index of the Alternate Mode within the SVID as returned in response to a Discover Modes command. Example: 0 - first Mode entry, 1 - second mode entry	
47	iAlternateModeString[0]	1	Index	Index of string descriptor describing protocol. It is optional to support this string.	
...	
44+ (n * 4)	wSVID[n]	2	Number	Standard or Vendor ID. This shall match one of the SVIDs returned in response to a USB PD Discover SVIDs command. If there is more than one Alternate Mode in the same SVID, the wSVID[n] should be repeated for each Alternate Mode.	
46 + (n * 4)	bAlternateMode[n]	1	Number	Index of the Alternate Mode within the SVID as returned in response to a Discover Modes command	
47 + (n * 4)	iAlternateModeString[n]	1	Index	Index of string descriptor describing protocol. It is optional to support this string.	

3.1.6.2.1 Example usage of wSVID[n], bAlternateMode[n], iAlternateModeString[n]:

In this example, the USB PD Discover SVIDs Command returns the Intel VID 0x8087 and the DP_SID 0xFF01.

Two Alternate Modes are returned for the Intel VID 0x8087. bAlternateMode[0] contains Mode 1 (Fictional Application) and bAlternateMode[1] contains Mode 2 (Fictional Debug). These are Intel Corporation proprietary Alternate Modes.

One Alternate Mode is returned for the DP_SID 0xFF01. bAlternateMode[2] contains Mode 1 for the DisplayPort SID. Details on interpretation of the Mode VDO response are available in the VESA DisplayPort Alt Mode on USB Type-C Standard.

Table 3-8 Example Values for wSVID, bAlternateMode, iAlternateModeString

wSVID[0]	0x8087	Intel VID
bAlternateMode[0]	0x00	Mode1 in list of modes (Fictional Application)
iAlternateModeString[0]	0x00	Index to a string describing the Fictional Application Alternate Mode.
wSVID[1]	0x8087	Intel VID
bAlternateMode[1]	0x01	Mode2 in list of modes (Fictional Debug)
iAlternateModeString[1]	0x01	Index to a string describing the Intel Fictional Debug Alternate Mode.
wSVID[2]	0xFF01	DP_SID
bAlternateMode[2]	0x00	Mode1 in list of modes (DisplayPort)

iAlternateModeString[2]	0x02	Index to a string describing the DisplayPort Alternate Mode.
--------------------------------	------	--

3.1.6.3 Billboard Alternate Mode Capability Descriptor

This descriptor is used in addition to the Billboard capability descriptor to describe additional properties for a given alternate mode. There shall be one such descriptor for each alternate mode.

Table 3-9: Billboard Alternate Mode Capability Descriptor

Offset	Field	Bytes	Value	Description
0	bLength	1	Number	Size of this Descriptor in bytes.
1	bDescriptorType	1	Constant	Descriptor type: DEVICE CAPABILITY Type.
2	bDevCapabilityType	1	Number	BILLBOARD ALTERNATE MODE CAPABILITY
3	bIndex	1	Number	Index at which the Alternate Mode appears in the array of Alternate Modes described in the Billboard Capability Descriptor
4	dwAlternateModeVdo	4	Number	Contents of the Mode VDO for the alternate mode identified by bIndex. Refer to the Standard or Vendor specification for details on information contained in the Mode VDO.

3.1.6.3.1 Example usage of Billboard Alternate Mode Capability Descriptor:

In this example, the USB PD Discover SVIDs Command returns the Intel VID 0x8087 and the DP_SID 0xFF01.

Two Alternate Modes are returned for the Intel VID 0x8087. These modes will each be described by an alternate mode capability descriptor. For the first alternate mode capability descriptor, dwAlternateModeVdo contains Mode VDO 1 (Fictional Application) and for the second alternate mode capability descriptor, dwAlternateModeVdo contains Mode VDO 2 (Fictional Debug). These are Intel Corporation proprietary Alternate Modes. These two alternate modes are located at index 0 and 1 in the Billboard descriptor.

One Alternate Mode is returned for the DP_SID 0xFF01. dwAlternateModeVdo for the corresponding descriptor contains Mode 1 VDO. Details on interpretation of the Mode VDO response are available in the VESA DisplayPort Alt Mode on USB Type-C Standard. This alternate mode is located at index 2 in the Billboard descriptor.

This will result in three separate Billboard Alternate Mode Capability Descriptors in series. These descriptors should follow the Billboard Capability Descriptor. Each Billboard Alternate Mode Descriptor is 8 bytes long. Adding the three descriptors increases the BOS Descriptor wTotalLength by 24 and bNumDeviceCaps by 3.

Table 3-10 Example Billboard Alternate Mode Capability Descriptors

bLength	8	Size of this Descriptor in bytes
bDescriptorType	DEVICE CAPABILITY	Descriptor type
bDevCapabilityType	BILLBOARD ALTERNATE MODE CAPABILITY	Device Capability Type
bIndex	0x00	Location of Intel Fictional Application Alternate Mode in the Billboard Descriptor
dwAlternateModeVdo	0x00000010	Mode1 VDO (Fictional Application) - from USB PD Mode response
bLength	8	Size of this Descriptor in bytes.
bDescriptorType	DEVICE CAPABILITY	Descriptor type
bDevCapabilityType	BILLBOARD ALTERNATE MODE CAPABILITY	Device Capability Type
bIndex	0x01	Location of Intel Fictional Debug Alternate Mode in the Billboard Descriptor

dwAlternateModeVdo	0x00000002	Mode2 VDO (Fictional Debug) – from USB PD Mode response
bLength	8	Size of this Descriptor in bytes.
bDescriptorType	DEVICE CAPABILITY	Descriptor type
bDevCapabilityType	BILLBOARD ALTERNATE MODE CAPABILITY	Device Capability Type
bIndex	0x02	Location of DisplayPort Mode1 Alternate Mode in the Billboard Descriptor
dwAlternateModeVdo	0x000C00C5	Mode1 VDO – from USB PD Mode response Refer to the VESA DP Alt Mode on USB Type-C Standard for details

4 Requests and Control Sequences

4.1 Standard Requests

The Billboard Device Class shall support at least the following standard requests described in Section 9, “USB Device Framework” of the USB Specification. The Billboard Device Class places no specific requirements on the values for the standard requests:

- Get Configuration
- Get Descriptor
- Get Status
- Set Address
- Set Configuration

4.2 Class-Specific Requests

This class does not support any class specific requests.

A Values of Constants

A.1. Billboard Class Code

Table 4-1 Billboard Class Code

Billboard Class Code	Value
BILLBOARD_CLASS	0x11

A.2. Billboard Subclass Codes

Table 4-2 Billboard Subclass Code

Billboard SubClass Code	Value
BILLBOARD_SUBCLASS	0x00

A.3. Billboard Protocol Codes

Table 4-3 Billboard Protocol Code

Billboard Protocol Code	Value
BILLBOARD_PROTOCOL	0x00

A.4. Billboard Descriptor Types

Table 4-4 Billboard Capability Descriptor Types

Billboard Capability Descriptor Type	Value
BILLBOARD	0x0D
BILLBOARD ALTERNATE MODE	0x0F

A.5. Parameter Values

Table 4-5 Parameter Values

CONSTANT	Minimum Value	Maximum Value
MAX_NUM_ALT_MODE	N/A	0x34