Connector Type Tests

Version 0. 73
November 13, 2018

Connector Type Tests

All Chapter 9 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 and configured. 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

Assertion #	Test Name	Assertion Description
9.1 Overview		
9.1#1	TD 9.1	All PD Capable USB (PDUSB) Devices shall report themselves as self-powered devices (over USB) when plugged into a PD capable port even if they are entirely powered from VBUS.
9.1#2	USB30CV/USB20CV current measurement tests	PDUSB Devices shall follow the PD requirements when it comes to suspend (see Section 6.4.1.3.2), configured, and operational power.
9.1#3	USB30CV/USB20CV current measurement tests	The power requirements reported in the PD Consumer Port descriptor of the device shall override the power draw reported in the bMaxPower field in the configuration descriptor.
9.1#4	Type-C Functional/PD TD 9.1 (temporary)	A PDUSB Device shall report zero in the bMaxPower field after successfully negotiating a mutually agreeable power contract.
9.1#5	Type-C Functional	A PDUSB Device shall disconnect and re-enumerate when it switches operation back to operating in standard [USB2.0], [USB3.0] or [BC1.2].
9.1#6	Type-C Functional/ PD TD 9.1 (temporary)	When operating in [USB2.0], [USB3.0] or [BC1.2] mode a PDUSB device shall report its power draw via the bMaxPower field.
9.1.1 PDUSB Device and Hub Requirements		
9.1.1#1	TD 9.2 to TD 9.7	All PDUSB Devices with data lines shall return all relevant PD descriptors.
9.1.1#2	TD 9.9	PDUSB Hubs shall also support a PD bridge as defined in [USBTypeCBridge 1.0].
9.1.2 Mapping to USB Device States		
9.1.2#1	not tested	The device shall determine whether or not it is in the USB Attached or USB Powered states as described in Figure 9-3, Figure 9-4 and Figure 9-5.
9.1.2#2	existing CV tests	All other USB states of the PDUSB Device shall be as described in Chapter 9 of the USB 3.0/USB 2.0 Specification.
9.1.3 PD Software Stack		
9.1.4 PDUSB Device Enumeration		
9.1.4#1		If a PDUSB Device cannot perform its intended function with the amount of power that it can get from the Port it is connected to then the host system Should display a Message (on a PD aware OS) about the failure to provide sufficient power to the device.
9.2 PD Specific Descriptors		
9.2#1	not tested	A PDUSB Device shall return all relevant descriptors mentioned in this section.

9.2#2	TD 9.2- TD 9.5	A PDUSB device shall return its capability descriptors as part of the device’s Binary Object Store (BOS) descriptor set.
9.2.1 USB Power Delivery Capability Descriptor		
9.2.1#1	TD 9.2	The bReserved field in USB Power Delivery Capability Descriptor shall be set to zero.
9.2.1#2	TD 9.2	The Reserved bits (0, 7, 31:16) of bmAttributes in USB Power Delivery Capability Descriptor shall be set to zero.
9.2.1#3	TD 9.2	The Battery Charging bit of bmAttributes in USB Power Delivery Capability Descriptor shall be set to one to indicate this device supports the Battery Charging Specification as per the value reported in the bcdBCVersion field.
9.2.1#4	TD 9.2	The USB Power Delivery bit of bmAttributes in USB Power Delivery Capability Descriptor shall be set to one if the device supports the USB Power Delivery Specification as per the value reported in the bcdPDVersion field.
9.2.1#5	TD 9.2	The Provider bit of bmAttributes in USB Power Delivery Capability Descriptor shall be set to one to indicate this device is capable of providing power.
9.2.1#6	TD 9.2	The Consumer bit of bmAttributes in USB Power Delivery Capability Descriptor shall be set to one to indicate that this device is a consumer of power.
9.2.1#7	TD 9.2	Bit 5 of bmAttributes in USB Power Delivery Capability Descriptor shall be set to 1 to indicate that this device supports the feature CHARGING_POLICY.
9.2.1#8	TD 9.2	The USB Type-C Current bit of bmAttributes in USB Power Delivery Capability Descriptor shall be set to one to indicate this device supports power capabilities defined in the USB Type-C specification as per the value reported in the bcdUSBTypeCVersion field.
9.2.1#9	TD 9.2	At least one of the bits 8, 9 or 14 of the bmPowerSource bits of USB Power Delivery Capability Descriptor shall be set to 1.
9.2.1#10	TD 9.2	The Reserved bit of bmPowerSource in USB Power Delivery Capability Descriptor shall be set to zero.
9.2.1#11	TD 9.2	If the PDUSB device supports an AC power supply, the AC Supply bit of bmAttributes in the USB Power Delivery Capability Descriptor shall be set to 1, else set to 0.
9.2.1#12	TD 9.2	If the PDUSB device supports a battery, the Battery bit of bmAttributes in the USB Power Delivery Capability Descriptor shall be set to 1, else 0.
9.2.1#13	TD 9.6	If the PDUSB device has a battery, the NumBatteries bits of bmAttributes in the USB Power Delivery Capability Descriptor shall report the number of batteries in the device.

9.2.1#14	TD 9.2	If the PDUSB devices uses Vbus, the Uses Vbus bit of bmAttributes in USB Power Delivery Capability Descriptor shall be set to 1, else 0.
9.2.1#15	TD 9.2	The bcdBCVersion field of the USB Power Delivery Capability Descriptor shall indicate the BC version in BCD supported by the device if the device indicates that it supports BC in the bmAttributes field.
9.2.1#16	TD 9.2	The bcdPDVersion field of the USB Power Delivery Capability Descriptor shall indicate the PD version in BCD supported by the device if the device indicates that it supports PD in the bmAttributes field.
9.2.1#17		The bcdUSBTypeCVersion field of the USB Power Delivery Capability Descriptor shall indicate the USB Type-C Specification Release Number in BCD if the device indicates it supports USB Type-C.
9.2.1#18	TD 9.2 (update)	The bLength field in the USB Power Delivery Capability Descriptor shall be set to 14d.
9.2.1#19	TD 9.2 (update)	The bDescriptorType field in the USB Power Delivery Capability Descriptor shall be set to 10h.
9.2.1#20	TD 9.2 (update)	The bDevCapabilityType field in the USB Power Delivery Capability Descriptor shall be set to 6h.
9.2.2 Battery Info Capability Descriptor		
9.2.2#1	TD 9.3	A PDUSB Device shall support the Battery Info Capability Descriptor if it reported that one of its power sources was a Battery in the bmPowerSource field in its Power Deliver Capability Descriptor.
9.2.2#2	TD 9.3	A PDUSB device shall return one Battery Info Descriptor per battery it supports.
9.2.2#3	TD 9.3	The iBattery field of the Battery Info Capability Descriptor shall contain the index of string descriptor that contains the user friendly name for this battery.
9.2.2#4	TD 9.3	The iSerial field of the Battery Info Capability Descriptor shall contain the index of the string descriptor that contains the Serial Number String for this battery.
9.2.2#5	TD 9.3	The iManufacturer field of the Battery Info Capability Descriptor shall contain the index of string descriptor that contains the name of the Manufacturer for this battery.
9.2.2#6	not tested	The bBatteryID field of the Battery Info Capability Descriptor shall be used to uniquely identify this battery in status messages.
9.2.2#7	TD 9.3	The bReserved field of the Battery Info Capability Descriptor shall be set to zero.
9.2.2#8	TD 9.3	The dwChargedThreshold of the Battery Info Capability Descriptor shall contain the Battery Charge value above which this battery is considered to be fully charged but not necessarily “topped off.”
9.2.2#9	TD 9.3	The dwWeakThreshold field of the Battery Info Capability Descriptor shall contain the minimum

		charge level of this battery such that above this threshold, a device can be assured of being able to power up successfully.
9.2.2#10	TD 9.3	The dwBatteryDesignCapacity field of the Battery Info Capability Descriptor shall contain the design capacity of the battery.
9.2.2#11	TD 9.3	The dwBatteryLastFullchargeCapacity field of the Battery Info Capability Descriptor shall contain the maximum capacity of the battery when fully charged.
9.2.2#12	TD 9.3	The bLength field in the Battery Info Capability Descriptor shall be set to 24.
9.2.2#13	TD 9.3	The bDescriptorType field in the Battery Info Capability Descriptor shall be set to 10h.
9.2.2#14	TD 9.3	The bDevCapabilityType field of the Battery Info Capability Descriptor shall be set to 7h.
9.2.3 PD Consumer Port Capability Descriptor		
9.2.3#1	TD 9.4	A PDUSB Device shall support the PD Consumer Port Capability Descriptor if it is a Consumer.
9.2.3#2	TD 9.4	The bReserved field of the PD Consumer Port Descriptor shall be set to zero.
9.2.3#3	TD 9.4	The bmCapabilities Battery Charging bit of the PD Consumer Port Descriptor shall be set if the consumer port will operate under Battery Charging.
9.2.3#4		The bmCapabilities USB Power Delivery bit of the PD Consumer Port Capability Descriptor shall be set if the consumer port will operate under PD.
9.2.3#5		The bmCapabilities USB Type-C Current bit of the PD Consumer Port Descriptor shall be set if the consumer port will operate under USB Type-C Current.
9.2.3#6		Bits 15:3 of the bmCapabilities of the PD Consumer Port Capability Descriptor are reserved and shall be set to zero.
9.2.3#7	operating current	The wMinVoltage field of the PD Consumer Port Descriptor shall contain the minimum voltage in 50mV units that this Consumer is capable of operating at.
9.2.3#8	operating current	The wMaxVoltage field of the PD Consumer Port Descriptor shall contain the maximum voltage in 50mV units that this Consumer is capable of operating at.
9.2.3#9	TD 9.4	The wReserved field of the PD Consumer Port Descriptor is reserved and shall be set to zero.
9.2.3#10	modify existing current measurement tests	The dwMaxOperatingPower field of the PD Consumer Port Descriptor shall contain the maximum power in 10mW units this Consumer can draw when it is in a steady state operating mode.

9.2.3#11	modify existing current measurement tests	The dwMaxPeakPower field of the PD Consumer Port Descriptor shall contain the maximum power in 10mW units this Consumer can draw for a short duration of time (dwMaxPeakPowerTime) before it falls back into a steady state.
9.2.3#12	not tested	The dwMaxPeakPowerTime field of the PD Consumer Port Descriptor shall contain the time in 100ms units that this Consumer can draw peak current.
9.2.3#13	not tested	The dwMaxPeakPowerTime field of the PD Consumer Port Descriptor shall be set to 0xFFFF if this value is unknown.
9.2.3#14	TD 9.4	The bLength field of the PD Consumer Port Descriptor shall be set to 24.
9.2.3#15	TD 9.4	The bDescriptorType field of the PD Consumer Port Descriptor shall be set to 10h.
9.2.3#16	TD 9.4	The bDevCapability field of the PD Port Capability Descriptor shall be set to 8h
9.2.4 PD Provider Port Capability Descriptor		
9.2.4#1	TD 9.5	A PDUSB Device shall support the PD Provide Port Capability Descriptor if it is a provider.
9.2.4#2	TD 9.5	The bReserved fields of the PD Provider Port Capability Descriptor is reserved and shall be set to zero.
9.2.4#3	TD 9.5	The bmCapabilities Battery Charging bit of the PD Provider Port Capability Descriptor shall be set if the provider port will operate under Battery Charging.
9.2.4#4		The bmCapabilities USB Power Delivery bit of the PD Provider Port Capability Descriptor shall be set if the provider port will operate under PD.
9.2.4#5		The bmCapabilities USB Type-C Current bit of the PD Provider Port Descriptor shall be set if the provider port will operate under USB Type-C Current.
9.2.4#6		Bits 15:3 of the bmCapabilities of the PD Provider Port Capability Descriptor are reserved and shall be set to zero.
9.2.4#7	TD 9.5	The bNumOfPDObjects field of the PD Provider Port Descriptor shall indicate the number of Power Data Objects.
9.2.4#8	TD 9.5	The PD Provider Port Descriptor shall contain bNumOfPDObjects Power Data Objects starting at wPowerDataObject1 field.
9.2.4#9	TD 9.5	All Power Data Objects contained in a PD Provider Port Descriptor shall be 4 bytes long.
9.2.4#10	TD 9.5	The bLength field of the PD Provider Port Descriptor shall be set to 8.
9.2.4#11	TD 9.5	The bDescriptorType field of the PD Provider Port Descriptor shall be set to 10h
9.2.4#12	TD 9.5	The bDevCapabilityType field of the PD Provider Port Descriptor shall be set to 9h.

9.2.4#13	**TD** 9.5	**All** Power Data Objects shall be valid Source PDOs as defined in Section 6.4.1.2 of the Power Delivery 3.0 Specification.
9.3 PD **Specific Requests and Events**		
9.3#1	n/a	A PDUSB Device that is compliant to the PD 3.0 specification shall support the battery related requests if it has a battery.
9.3#2		**A** PDUSB Hub that is compliant the PD 3.0 specification shall support a USB PD Bridge.
9.3.1 PD **Specific Requests**		
9.3.1#1		The PD class defines requests to which PDUSB Devices shall respond as outlined in Table 9-6.
9.3.1#2	TD 9.7	All valid requests in Table 9-6 shall be implemented by PDUSB devices.
9.4 PDUSB Hub and PDUSB Peripheral Device Requests		
9.4.1 GetBatteryStatus		
9.4.1#1		The GetBatteryStatus request returns the current status of the Battery in a PDUSB Hub/Peripheral.
9.4.1#2	TD 9.6	The PDUSB Hub/Peripheral Device shall return the Battery Status of the Battery identified by the value of wIndex field in a Get Battery Status request.
9.4.1#3	TD 9.6	Every PDUSB Device that has a battery shall return its Battery Status when queried with a GetBatteryStatus request.
9.4.1#4	TD 9.6	A GetBatteryStatus with wValue of 0 and wIndex of valid Battery ID shall return 8 bytes of data.
9.4.1#5	TD 9.6	For Providers or Consumers with multiple batteries, the status of each battery shall be reported per battery.
9.4.1#6	TD 9.6	The bBatteryAttributes field of data returned from GetBatteryStatus request shall be set to 0 if there is no battery currently attached.
9.4.1#7	TD 9.6	The bBatteryAttributes of data returned from GetBatteryStatus request shall not have a value greater than 3.
9.4.1#8	TD 9.6	The bBatterySOC field of data returned from GetBatteryStatus request shall indicate the Battery State of Charge given as percentage value from Battery Remaining Capacity.
9.4.1#9	TD 9.6	The bBatteryStatus field of data returned from GetBatteryStatus request shall indicate the present status of the battery as described in table 9-9 of USB PD spec.
9.4.1#10	TD 9.6	The bBatteryStatus field of data returned from GetBatteryStatus request shall not be greater than 7.
9.4.1#11	TD 9.7	The bRemoteWakeCapStatus field of data returned from GetBatteryStatus request shall indicate if the

		device is enabled for Battery Remote wake events as described in Table 9-9 of USB PD Spec.
9.4.1#12	TD 9.7	If the device supports remote wake, it shall support Battery remote wake events.
9.4.1#13	TD 9.6	The default value for the Remote wake events shall be turned off (set to zero).
9.4.1#14		If (bRemoteWakeCapStatus bits are) set to one the device shall generate a wake event when a change of status occurs.
9.4.1#15	TD 9.6	Bits 7:3 of the bRemoteWakeCapStatus are Reserved and shall be set to zero.
9.4.1#16	not tested	The wRemainingOperatingTime field of data returned from GetBatteryStatus request shall contain the operating time (in minutes) until the Weak Battery threshold is reached.
9.4.1#17	not tested	The wRemainingOperatingTime field of GetBatteryStatus request shall exclude any additional power received from charging.
9.4.1#18	not tested	The wRemainingOperatingTime field of data returned from GetBatteryStatus request shall return a value of 0xFFFF if the battery is not capable of returning this information.
9.4.1#19	not tested	The wRemainingChargeTime field of data returned from GetBatteryStatus shall contain the remaining time (in minutes) until the Charged Battery threshold is reached.
9.4.1#20	not tested	The wRemainingChargeTime field of data returned from GetBatteryStatus request shall only be valid if the Charging Flow is “Charging”.
9.4.1#21	not tested	The wRemainingChargeTime field of data returned from GetBatteryStatus request shall return a value of 0xFFFF if the battery is not capable of returning this information.
9.4.1#22	not tested	If wIndex refers to a Battery that does not exist, then the PDUSB Device shall respond with a Request Error.
9.4.1#23	TD 9.6	The bBatteryAttributes field of data returned from GetBatteryStatus request shall be set to 1 if the battery is charging.
9.4.1#24	TD 9.6	The bBatteryAttributes field of data returned from GetBatteryStatus request shall be set to 2 if the battery is discharging.
9.4.1#25	**TD** 9.6	**The** bBatteryAttributes field of data returned from GetBatteryStatus request shall be set to 3 if the battery is neither discharging nor charging.
9.4.2 SetPDFeature		
9.4.2.1 BATTERY_**WAKE_MASK Feature Selector**		
9.4.2.1#1	TD 9.7	When the Battery Present bit is set in a SetPDFeature() request, then the PDUSB Device shall

		generate a wake event if it detects that a battery has been attached.
9.4.2.1#2	not tested	When the Charging Flow bit is set in a SetPDFeature() request then the PDUSB Device shall generate a wake event if it detects that a battery switched from charging to discharging or vice versa.
9.4.2.1#3	TD 9.7	When the Battery Error bit is set in a SetPDFeature() request then the PDUSB Device shall generate a wake event if the battery has detected an error condition.
9.4.2.1#4	not tested	A PDUSB device shall respond with a Request Error if wValue is not a feature selector listed in Table 9 7
9.4.2.1#5	**not** tested	**A** PDUSB device shall respond with a Request Error if wIndex of SetPDFeature() request is not set to a value as specified above.
9.4.2.2 CHARGING_**POLICY Feature Selector**		
9.4.2.2#1	not tested	If the device is using USB Type-C Current above the default value or is using PD then the CHARGING_POLICY feature setting has no effect and the rules for power levels specified in the USB Type-C 1.2 or USB PD specifications shall apply.
9.4.2.2#2	TD 9.8	If the wIndex of CHARGING_POLICY feature selector is set to 00h, the device shall follow the default current limits as defined in the USB 2.0 or USB 3.1 specification, as negotiated through other USB mechanisms such as BC.
9.4.2.2#3		The value 00h is the default wIndex value for the CHARGING_POLICY feature selector.
9.4.2.2#4		If the wIndex of CHARGING_POLICY feature selector is set to 01h, the device may draw additional power during the unconfigured and suspend states for the purposes of charging.
9.4.2.2#5		If the wIndex of CHARGING_POLICY feature selector is set to 01h, the device shall limit its current draw to the higher of these two values: ICCHPF as defined in USB 2.0/3.1 and current limit as negotiated.
9.4.2.2#6		If the wIndex of CHARGING_POLICY feature selector is set to 02h, the device may draw additional power during the unconfigured and suspend states for the purposes of charging.
9.4.2.2#7		If the wIndex of CHARGING_POLICY feature selector is set to 02h, the device shall limit its current draw to the higher of these two values: ICCHPF as defined in USB 2.0/3.1 and current limit as negotiated.
9.4.2.2#8		If the wIndex of CHARGING_POLICY is set to 03h, the device shall not consume any current for charging itself regardless of its USB state.

9.4.2.2#9		The CHARGING_POLICY Feature Selector command is valid for the PDUSB Hub/Peripheral in the Address or Configured USB states.
9.4.2.2#10		The device will go back to the wIndex default value of 0 for feature selector CHARGING_POLICY whenever it is reset.

Compliance Tests

9.1 Configuration Descriptor Test

Purpose	Verify Configuration Descriptor
Critical for Safety	No
Applies to	PDUSB peripherals
Description	Query DUT for configuration descriptor and check pertinent PD fields.
Test setup	USB30CV
Preconditions	
Assertion References	9.1#1, 9.1#4
Device States Tested	Default, Addressed, Configured
Power States Tested	Standard USB Power, USB Type-C™ Power, PD Power Contract
Checklist References	

Test Procedure

For each power state supported by the device, run the following test steps.

1. Enumerate the DUT and put in required device state.
   For each configuration supported by the device:
2. Send a GetDescriptor(wValue, wIndex, wLength, Data) request with the following values:
   - wValue set to 2 (Configuration) | Configuration Index
   - wIndex set to 0
   - wLength set to 9d

   Test fails if the device does not return the descriptor or if the length returned is not 9d.

3. Send a GetDescriptor(wValue, wIndex, wLength, Data) request with the following values:
   - wValue set to 2 (Configuration) | Configuration Index
   - windex set to 0
   - wLength set to wTotalLength (returned in previous call to GetDescriptor)

   Test fails if the device does not return the descriptor or if the length is not wTotalLength.

If DUT is connected via USB-Type-C™ Current or Power Delivery Contract

4. The DUT must report self-powered.
5. Verify that ConfigurationDescriptor.bmAttributes bit 6 (Self-Powered) is 1. (9.1#1).
6. Verify that ConfigurationDescriptor.bMaxPower field as 0. (9.1#4)

9.2 Power Delivery Capability Descriptor Test

Purpose	Verify Power Delivery Capability Descriptor
Critical for Safety	No
Applies to	PDUSB peripherals
Description	Query DUT for Power Delivery Capability Descriptor and verify fields.
Test setup	USB30CV

Preconditions	
Assertion References	9.1.1#1, 9.2#2, 9.2.1#1-9.2.1#20, 9.2.1#17
Device States Tested	Default, Addressed, Configured
Checklist References	
Assumptions *(this entry to be taken out when the base specs are definitively clarified)*	• This descriptor will be present for all USB devices with a USB Type-C Upstream Facing Port, even if they don't support PD. • All entries in the Descriptor are static fields that stay the same regardless of the current connection type. • **bmAttributes** *Provider* and *Consumer* bits are valid even if the *USB Power Delivery* bit is not set. • **bmPowerSource** *Uses Vbus* must be set to 1. • **bmAttributes** *USB Type Current* bit is only set when the UFP supports current above the USB 2 and USB 3 defaults.

Test Procedure

1. Enumerate the DUT and put in required device state.
2. Send a GetDescriptor(wValue, wIndex, wLength, Data) request with the following values:
    - wValue set to 15d (BOS)
    - wIndex set to 0
    - wLength set to 5d (All of the configuration Descriptor)

    Test fails if the device does not return the descriptor or if the length returned is not 5d. (9.1.1#1)

3. Send a GetDescriptor(wValue, wIndex, wLength, Data) request with the following values:
    - wValue set to 15d (BOS)
    - windex set to 0
    - wLength set to wTotalLength (returned in previous call to GetDescriptor)

    Test fails if the device does not return the descriptor or if the length is not wTotalLength. (9.1.1#1, 9.2#2)

4. Parse the data, keeping the BOS Power Delivery Capability Descriptor. Perform the remaining steps on the BOS Power Delivery Capability Descriptor.
5. Test fails if there is not exactly 1 BOS Power Delivery Capability Descriptor contained in the BOS descriptor set. (9.2#2)
6. Test fails if bLength is not 14d. (9.2.1#18)
7. Test fails if bDescriptorType is not 10h. (9.2.1#19)
8. Test fails if bDevCapabilityType is not 6h. (9.2.1#20)
9. Test fails if bReserved is not 0. (9.2.1#1)
10. Test fails if bmAttributes bits 0, 7, 31:16 are not 0. (9.2.1#2)
11. Test fails if AC Supply (bit 8), Battery (bit 9) and Uses VBus (bit 14) of bmAttributes are all 0. (9.2.1#9)
12. Test fails if Reserved bit of bmPowerSource is not 0. (9.2.1#10)
13. Test fails if Battery Charging bit is set and bcdBCVersion field does not equal 120h. *(no assert)*

14. Test fails if Power Delivery bit is set and bcdPDVersion field does not equal to one of 200h or 300h. *(no assert)*
15. Test fails if USB Type-Current bit is set and bcdUSBTypeCVersion does not equal 130h. *(no assert)*

VIF Checks:

1. Read VIF file for DUT UFP

If VIF field is equal to this	**bmAttributes field must be this**	**Assertion**
BC_1_2_Support = (“Portable Device” or “Both”)	Battery Charging = 1	9.2.1#3
BC_1_2_Support != (“Portable Device” or “Both”)	Battery Charging =0	9.2.1#3
USB_PD_Support = N	USB Power Delivery = 0	9.2.1#4
USB_PD_Support = Y	USB Power Delivery = 1	9.2.1#4
Type_C_State_Machine = SRC or DRP	Provider = 1	9.2.1#5
Type_C_State_Machine = SNK or DRP	Consumer = 1	9.2.1#6
Connector_Type = 2 (Type-C) *(should always be the case)*	USB Type-C Current = 1	9.2.1#8
Unconstrained_Power = N	bmPowerSource.AC Supply = 0	9.2.1#11
Unconstrained_Power = Y	*Prompt user for bmPowerSource.AC Supply.*	9.2.1#11
Num_Fixed_Batteries + Num_Swappable_Battery_Slots = 0 *(if USB_PD_Support = N or PD_Specification_Revision is not PD 3.0, then prompt user for total battery count)*	bmPowerSource.Battery = 0	9.2.1#12
Num_Fixed_Batteries + Num_Swappable_Battery_Slots > 0 *(if USB_PD_Support = N or PD_Specification_Revision is not PD 3.0, then prompt user for total battery count)*	bmPowerSource.Battery = 1	9.2.1#12
Num_Fixed_Batteries + Num_Swappable_Battery_Slots *(if USB_PD_Support = N or PD_Specification_Revision is not PD 3.0, then prompt user for total battery count)*	bmPowerSource.NumBatteries = Num_Fixed_Batteries + Num_Swappable_Battery_Slots	9.2.1#13
BC_1_2_Support = (“Portable Device” or “Both”)	bcdBCVersion = 0x0120	9.2.1#15
PD_Specification_Revision = PD 2.0	bcdPDVersion = 0x0200	9.2.1#16
PD_Specification_Revision = PD 3.0	bcdPDVersion = 0x0300	9.2.1#16

2. Make sure Uses VBus field of bmAttributes is set to 1. (9.2.1#14)

9.3 Battery Info Capability Descriptor Test

Purpose	Verify Battery Info Capability Descriptor
Critical for Safety	No
Applies to	PDUSB peripherals
Description	Query DUT descriptors and check pertinent PD fields.
Test setup	USB30CV
Preconditions	
Assertion References	9.1.1#1, 9.2.1#8, 9.2.2#1, 9.2.2#2, 9.2.2#3, 9.2.2#4, 9.2.2#5, 9.2.2#7, 9.2.2#12, 9.2.2#13, 9.2.2#14, 9.2#2
Device States Tested	Default, Addressed, Configured
Checklist References	
Assumptions *(this entry to be taken out when the base specs are definitively clarified)*	• This descriptor will be present for all battery powered USB devices with a USB Type-C Upstream Facing Port, even if they don't support PD. • All entries in the Descriptor are static fields that stay the same regardless of the current connection type.

Test Procedure

1. Enumerate the DUT and put in required device state.
2. Send a GetDescriptor(wValue, wIndex, wLength, Data) request with the following values:
    - wValue set to 15d (BOS)
    - wIndex set to 0
    - wLength set to 5d (All of the configuration Descriptor)

    Test fails if the device does not return the descriptor or if the length returned is not 5d. (9.1.1#1)

3. Send a GetDescriptor(wValue, wIndex, wLength, Data) request with the following values:
    - wValue set to 15d (BOS)
    - windex set to 0
    - wLength set to wTotalLength (returned in previous call to GetDescriptor)

    Test fails if the device does not return the descriptor or if the length is not wTotalLength. (9.1.1#1, 9.2#2)

4. Check the size of the returned descriptor table. Test fails if the length of the returned descriptor table is not wTotalLength.
5. Parse the data, keeping the BOS Power Delivery Capability Descriptor and Battery Info Capability Descriptor(s).
6. Test fails if bLength is not 24d. (9.2.2#12)
7. Test fails if bDescriptorType is not 10h. (9.2.2#13)
8. Test fails if bDevCapabilityType is not 7h. (9.2.2#14)
9. If bmPowerSource field bit 9 (Battery) in the PD Capability Descriptor is 0, fail the test if the device returned a Battery Info Capability Descriptor. (9.2.2#1)
10. If bmPowerSource field bit 9 (Battery) in the PD Capability Descriptor is 1, fail the test if the device did not return a Battery Info Capability Descriptor (9.2.2#1)

11. Test fails if the number of Battery Info Capability Descriptors does not match bmPowerSource field bits 11-13 (NumBatteries) in PD Capability Descriptor. (9.2.2#2)
12. Validate the string descriptor referenced by iBattery. (9.2.2#3)
13. Print the iBattery string descriptor.
14. Validate the string descriptor referenced by iSerial is present. (9.2.2#4)
15. Print the iSerial string descriptor.
16. Validate the string descriptor referenced by iManufacturer. (9.2.2#5)
17. Print the iManufacturer string descriptor.
18. Print bBatteryID value.
19. Test fails if bReserved is not 0. (9.2.2#7)
20. Print dwChargedThreshold, dwWeakThreshold, dwBatteryDesignCapacity and dwBatterylastFullChargeCapacity.
21. Test fails if any bBatteryID appears in more than one Battery Info Capability Descriptor. (9.2.2#6)

9.4 PD Consumer Port Capability Descriptor Test

Purpose	Verify Consumer Port Capability Descriptors
Critical for Safety	No
Applies to	PDUSB peripherals
Description	Query DUT descriptors and check pertinent PD fields.
Test setup	USB30CV
Preconditions	
Assertion References	9.1.1#1, 9.2.3#1-9.2.3#6, 9.2.3#9, 9.2.3#14-9.2.3#16, 9.2#2
Device States Tested	Default, Addressed, Configured
Checklist References	
Assumptions *(this entry to be taken out when the base specs are definitively clarified)*	• This descriptor will be present for all USB devices with a USB Type-C Upstream Facing Port, even if they don't support PD. • All entries in the Descriptor are static fields that stay the same regardless of the current connection type. • **bmCapabilities** is a bitmap (meaning multiple bits may be set). • **bmCapabilities** *USB Type-C Current* is only set if the UFP supports current above USB 2/USB 3 defaults. • There is no way to test **dwMaxOperatingPower**, **dwMaxPeakPower**, and **dwMaxPeakPowerTime**, as they do not appear to reference anything in the PD 3 spec.

Test Procedure

1. Enumerate the DUT and put I required device state.
2. Send a GetDescriptor(wValue, wIndex, wLength, Data) request with the following values:
    a. wValue set to 15d (BOS)
    b. wIndex set to 0
    c. wLength set to 5d (All of the configuration Descriptor)

    Test fails if the device does not return the descriptor or if the length returned is not 5d. (9.1.1#1)

3. Send a GetDescriptor(wValue, wIndex, wLength, Data) request with the following values:
    a. wValue set to 15d (BOS)
    b. wIndex set to 0
    c. wLength set to wTotalLength (returned in previous call to GetDescriptor)

    Test fails if the device does not return the descriptor or if the length is not wTotalLength. (9.1.1#1, 9.2#2)

4. Check the size of the returned descriptor table. Test fails if the length of the returned descriptor table is not wTotalLength.
5. Parse the data, keeping the BOS Power Delivery Capability Descriptor and Power Delivery Consumer Port Capability Descriptor.
6. Test fails if the USB Power Delivery Capability Descriptor consumer bit is set and there is not exactly 1 PD Consumer Port Capability Descriptor. (9.2.3#1)
7. Test fails if the USB Power Delivery Capability Descriptor Consumer bit is reset and there is a PD Consumer Port Capability Descriptor. (9.2.3#1)
8. Test fails if the bReserved field of the PD Consumer Port Descriptor is not zero. (9.2.3#2)
9. Test fails if bits 15:3 of the bmCapabilities of the PD Consumer Port Capability Descriptor are not set to zero. (9.2.3#6)
10. Test fails if the wReserved field is not zero. (9.2.3#9)
11. Test fails if the bLength field is not set to 24. (9.2.3#14)
12. Test fails if the bDescriptorType is not set to 10h. (9.2.3#15)
13. Test fails if the bDevCapabilityField is not set to 8h. (9.2.3#16)

VIF Checks:

1. Read VIF file for DUT UFP

If VIF field is equal to this	Descriptor field should be equal to	Assertion
The minimum value of all the Snk_PDO_Voltage (Snk_PDO_Supply_Type = Fixed) and Snk_PDO_Min_Voltage (Snk_PDO_Supply_Type != Fixed) fields across all Sink PDOs. *(Or prompt user if DUT doesn't support PD)*	< wMinVoltage	
The minimum value of all the Snk_PDO_Voltage (Snk_PDO_Supply_Type = Fixed) and Snk_PDO_Max_Voltage (Snk_PDO_Supply_Type != Fixed) fields across all Sink PDOs. *(Or prompt user if DUT doesn't support PD)*	> wMaxVoltage	

9.5 PD Provider Port Capability Descriptor Test

Purpose	Verify Provider Port Capability Descriptor
Critical for Safety	No
Applies to	PDUSB peripherals

Description	Query DUT descriptors and check pertinent PD fields.
Test setup	USB30CV
Preconditions	
Assertion References	9.1.1#1, 9.2.4#1, 9.2.4#2, 9.2.4#3, 9.2.4#4, 9.2.4#5, 9.2.4#6, 9.2.4#7, 9.2.4#8, 9.2.4#9, 9.2.4#11-9.2.4#15
Device States Tested	Default, Addressed, Configured
Checklist References	
Assumptions *(this entry to be taken out when the base specs are definitively clarified)*	• This descriptor will only be present on PDUSB devices that can act as a PD Provider. • All entries in the Descriptor are static fields that stay the same regardless of the current connection type. • **bmCapabilities** is a bitmap (meaning multiple bits may be set). • **bmCapabilities** *USB Type-C Current* is only set if the UFP supports current above USB 2/USB 3 defaults. • In **wPowerDataObjectN[0]** *USB Suspend Supported* is static and shall not change depending on its current PD status, and corresponds (negatively) to the VIF field ***USB_Suspend_May_Be_Cleared***. *(Note: Per the UUT spec, this bit is dynamic, and a UUT may choose to change its value in different **Source_Capabilities** messages.)* • For all other fields in **wPowerDataObjectN[0]** and in all other Source PDOs in **wPowerDataObjectN**, the fields are static and do not change depending on current PD status.

Test Procedure

1. Enumerate the DUT and put in required device state.
2. Send a GetDescriptor(wValue, wIndex, wLength, Data) request with the following values:
   a. wValue set to 15d (BOS)
   b. wIndex set to 0
   c. wLength set to 5d (All of the configuration Descriptor)

   Test fails if the device does not return the descriptor or if the length returned is not 5d. (9.1.1#1**)**

3. Send a GetDescriptor(wValue, wIndex, wLength, Data) request with the following values:
   a. wValue set to 15d (BOS)
   b. windex set to 0
   c. wLength set to wTotalLength (returned in previous call to GetDescriptor)

   Test fails if the device does not return the descriptor or if the length is not wTotalLength. (9.1.1#1)

4. Check the size of the returned descriptor table. Test fails if the length of the returned descriptor table is not wTotalLength.
5. Parse the data, keeping the BOS Power Delivery Capability Descriptor and Power Delivery Provider Port Capability Descriptor.
6. Test fails if the USB Power Delivery Capability Descriptor consumer bit is set and there is not exactly 1 PD Provider Port Capability Descriptor. (9.2.4#1)

7. Test fails if the USB Power Delivery Capability Descriptor Consumer bit is reset and there is a PD Provider Port Capability Descriptor. (9.2.4#1)
8. Test fails if the bReserved field at offset 3 of the PD Provider Port Descriptor is not zero. (9.2.4#2)
9. Test fails if the bReserved field at offset 7 of the PD Provider Port Descriptor is not zero. (9.2.4#2)
10. Test fails if bits 15:3 of the bmCapabilties of the PD Provider Port Capability Descriptor are not set to zero. (9.2.4#6)
11. Test fails if there are not bNumbOfPDOObjects Power Data Objects starting at wPowerDataObject1 field. (9.2.4#8)
12. Test fails if all Power Data Objects are not 4 bytes long (9.2.4#9)
13. Test fails if the bLength field is not set to 8. (9.2.4#10)
14. Test fails if the bDescriptorType is not set to 10h. (9.2.4#11)
15. Test fails if the bDevCapabilityField is not set to 9h. (9.2.4#12)

VIF checking:

VIF Checks:

1. Read VIF file for DUT UFP

If VIF field is equal to this	Descriptor field should be equal to	Assertion
Num_SrcPDOs =	bNumOfPDObjects	9.2.4#7
For each PDO in VIF	**PDO in Descriptor should match** (see section 6.4.1.2 of PD 3 spec)	9.2.4#8
Src_PDO_Supply_Type =	bits 31:30 of PDO	9.2.4#8
For each PDO with Src_PDO_Supply_Type = Fixed		
Src_PDO_Peak_Current =	bits 21:20 of PDO	9.2.4#8
Src_PDO_Voltage =	bits 19:10 of PDO	9.2.4#8
Src_PDO_Max_Current =	bits 9:0 of PDO	9.2.4#8
For each PDO with Src_PDO_Supply_Type = Variable		
Src_PDO_Min_Voltage =	bits 19:10 of PDO	9.2.4#8
Src_PDO_Max_Voltage =	bits 29:20 of PDO	9.2.4#8
Src_PDO_Max_Current =	bits 9:0 of PDO	9.2.4#8
For each PDO with Src_PDO_Supply_Type = Battery		
Src_PDO_Min_Voltage =	bits 19:10 of PDO	9.2.4#8
Src_PDO_Max_Voltage =	bits 29:20 of PDO	9.2.4#8
Src_PDO_Max_Power =	bits 9:0 of PDO	9.2.4#8
For each APDO with Src_PDO_Supply_Type = PPS		
Src_PDO_Min_Voltage =	bits 15:8 of PDO	9.2.4#8
Src_PDO_Max_Voltage =	bits 24:17 of PDO	9.2.4#8
Src_PDO_Max_Current =	bits 6:0 of PDO	9.2.4#8

9.6 Battery Status Test

Purpose	Verify Battery Status reporting
Critical for Safety	No
Applies to	PDUSB peripherals
Description	Query DUT for battery descriptors and check values if DUT has batteries.
Test setup	USB30CV
Preconditions	
Assertion References	9.1.1#1, 9.2.2#1, 9.2.1#11, 9.2.2#2, 9.4.2#1, 9.4.2#2, 9.4.2#3, 9.4.2#4, 9.4.2#5, 9.4.2#21, 9.4.2#22, 9.4.2#23, 9.4.2#9, 9.4.2#12, 9.4.2#13, 9.4.2#8, 9.2#2
Device States Tested	Configured
Checklist References	

Test Procedure

1. Prompt the user to connect the DUT with all batteries attached (if DUT has batteries).
2. Send a GetDescriptor(wValue, wIndex, wLength, Data) request with the following values:
    - wValue set to 15d (BOS)
    - wIndex set to 0
    - wLength set to 5d (All of the configuration Descriptor)
3. Test fails if the device does not return the descriptor or if the length returned is not 5d. (9.1.1#1)
4. Send a GetDescriptor(wValue, wIndex, wLength, Data) request with the following values:
    - wValue set to 15d (BOS)
    - windex set to 0
    - wLength set to wTotalLength (returned in previous call to GetDescriptor)
5. Test fails if the device does not return the descriptor or if the length is not wTotalLength. (9.1.1#1, 9.2#2)
6. Parse the data, keeping the BOS Power Delivery Capability Descriptor and Battery Info Capability Descriptor(s).
7. If bit 9 of USB Power Delivery Capability Descriptor is 0, device does not have a battery. Steps for device that does not have a battery:
    a. If device returned any Battery Info Capability Descriptors, fail test. (9.2.2#1)
    b. Skip remaining steps in this test.
8. Query VIF for number of batteries supported.
9. Verify that the number of batteries reported matches Battery info Capability Descriptor bmPowerSource NumBatteries (bits 11:13). (9.2.1#12)
10. Verify that the number of batteries reported by the user matches the number of Battery Info Descriptors returned by the DUT. (9.2.2#2)

Test Case for Attached Battery:

11. For each Battery Info Descriptor:
    a. Issue GetBatteryStatus request
        i. bRequest set to 21 (GET_BATTERY_STATUS)

        - ii. wValue set to 0
        - iii. wIndex set to bBatteryID from Battery Info Descriptor
        - iv. wLength set to 8
    - b. Test fails if device does not return 8 bytes. (9.4.1#1, 9.4.1#2, 9.4.1#3)
    - c. Test fails if bBatteryAttributes (byte 0) of returned data is 0, indicating no battery present. (9.4.1#5)
    - d. Test fails if bBatteryAttributes (byte 0) returns a value greater than 3. (9.4.2#7)
    - e. Print out Battery SOC (byte 1), Remaining Operating time (bytes 4 and 5) and Remaining Charge time (bytes 6 and 7).
    - f. Test fails if byte 2 (bBatteryStatus) of returned data is greater than 7. (9.4.2#10)
    - g. Test fails if byte 3 (bRemoteWakeCapStatus) of returned data is non-zero. (9.4.2#13, 9.4.2#14)

Test Case for Detached Battery:

12. For all Battery IDs:
    - a. Prompt user to remove battery associated with Battery ID being tested.
    - b. Issue GetBatteryStatus request
        - i. bRequest set to 21 (GET_BATTERY_STATUS)
        - ii. wValue set to 0
        - iii. wIndex set to bBattery ID from Battery Info Descriptor
        - iv. wLength set to 8
    - c. Test fails if device does not return 8 bytes. (9.4.2#1, 9.4.2#3, 9.4.2#4)
    - d. Test fails if bBatteryAttributes field (byte 0) of returned data is not 0. (9.4.2#5)
    - e. Print out Battery SOC (byte 1), Remaining Operating time (bytes 4 and 5) and Remaining Charge time (bytes 6 and 7).
    - f. Test fails if byte 2 (bBatteryStatus) of returned data is greater than 2. (9.4.2#8)
    - g. Test fails if byte 3 (bRemoteWakeCapStatus) of returned data is non-zero. (9.4.2#12, 9.4.2#13)
    - h. Prompt user to re-attach battery.

9.7 Power Delivery Remote Wake Test

Purpose	Verify that appropriate battery events generate a USB wake.
Critical for Safety	No
Applies to	PDUSB peripherals that have batteries
Description	Enable DUT for wake on PD events and verify correct operation.
Test setup	USB30CV
Preconditions	
Assertion References	9.4.2#11, 9.4.5#1, 9.4.5#2, 9.4.5#3, 9.4.2#10, 9.4.2#11
Device States Tested	Configured
Checklist References	

Test Procedure

1. Enumerate and configure the DUT.
2. Determine if the DUT is remote wake capable:
    - a. If the device is LS, FS or HS:

        i. Read the configuration descriptor for the DUT.
        ii. If bit 5 of the bmAttributes field is set, this device supports Remote Wake.
    b. If the device is SuperSpeed (Gen x)
        i. Read the full configuration descriptor for the DUT.
        ii. For each interface contained in the configuration descriptor:
            1. Send GetStatus request to the interface.
            2. If bit 1 of the data returned from GetStatus is 1, this device supports Remote Wake.

3. If DUT does not support remote wake, end test.
4.
5. Determine if DUT has batteries:
    a. Send a GetDescriptor(wValue, wIndex, wLength, Data) request with the following values:
        i. wValue set to 15d (BOS)
        ii. wIndex set to 0
        iii. wLength set to 5d (All of the configuration Descriptor)

       Test fails if the device does not return the descriptor or if the length returned is not 5d. (9.1.1#1)

    b. Send a GetDescriptor(wValue, wIndex, wLength, Data) request with the following values:
        i. wValue set to 15d (BOS)
        ii. wIndex set to 0
        iii. wLength set to wTotalLength (returned in previous call to GetDescriptor)

       Test fails if the device does not return the descriptor or if the length is not wTotalLength. (9.1.1#1)

    c. Check the size of the returned descriptor table. Test fails if the length of the returned descriptor table is not wTotalLength.
    d. Parse the data, keeping the BOS Power Delivery Capability Descriptor and any Battery Info Capability Descriptors
14. Get number of device batteries from VIF.
6. If bit 9 of bmPowerSource of USB Power Delivery Capability Descriptor is 0, device does not have batteries. End test.
7. Issue Set PD Feature to DUT:
    a. bRequest set to SET_FEATURE
    b. wValue set to 40d (BATTERY_WAKE_MASK)
    c. wIndex set to 0x01 (Battery Present)
    d. Test fails if SetPDFeature fails.
8.

   For all Batteries:
9. Prompt user to remove battery.
10. Issue GetBatteryStatus for BatteryID.
11. Verify that bBatteryWakeCapStat is set to 1, Wake on Battery Present. (9.4.1#11)

12. Verify that bBatteryAttributes is 0.
13. Suspend the device.
    a. If device is LS/FS/HS:
        i. Issue SetPortFeature(PORT_SUSPEND) to Hub downstream port where DUT is attached.
        ii. Test fails if SetPortFeature call fails.
    b. If device is SS:
        i. Issue SetFeature(FUNCTION_SUSPEND) to all interfaces that support function suspend.
        ii. Test fails if any SetPortFeature call fails.
14. Issue GetPortStatus and wait until the port is suspended.
15. Prompt user to insert the DUT battery.
16. Check Port Status to make sure link returned to U0 or L0 as appropriate. (9.4.2#11, 9.4.5#1)
17. Issue GetStatus() to device to ensure device is working. (9.4.2#11)
18. If device is SuperSpeed, ensure that a function remote wake event was received.
19. Issue GetBatteryStatus to device.
20. Verify bBatteryAttributes is 1, 2, or 3.

Charging:

21. Issue Set PD Feature to DUT:
    a. bRequest set to SET_FEATURE
    b. wValue set to 40d (BATTERY_WAKE_MASK)
    c. wIndex set to 0x02 (Charging Flow)
    d. Test fails if SetPDFeature fails
22. For all batteries
    a. Issue GetBatteryStatus for BatteryID
    b. Verify that bBatteryWakeCapStat is set to 0x02, Wake on Charging Flow. (9.4.1#11)

Battery Error

23. Issue Set PD Feature to DUT:
    a. bRequest set to SET_FEATURE
    b. wValue set to 40d (BATTERY_WAKE_MASK)
    c. wIndex set to 0x04 (Battery Error)
    d. Test fails if SetPDFeature fails.
24. For all batteries
    a. Issue GetBatteryStatus for BatteryID
    b. Verify bBatteryWakeCapStat is set to 0x04, Battery Error. (9.4.1#11)

9.8 Power Delivery CHARGING_POLICY Test

Purpose	Verify CHARGING_POLICY functionality
Critical for Safety	No
Applies to	PDUSB devices

Description	If DUT supports CHARGING_POLICY, make sure it supports the appropriate SetFeature calls.
Test setup	USB30CV
Preconditions	
Assertion References	
Device States Tested	Configured
Checklist References	

Test Procedure

1. Send a GetDescriptor(wValue, wIndex, wLength, Data) request with the following values:
    - wValue set to 15d (BOS)
    - wIndex set to 0
    - wLength set to 5d (All of the configuration Descriptor)
2. Test fails if the device does not return the descriptor or if the length returned is not 5d. (9.1.1#1)
3. Send a GetDescriptor(wValue, wIndex, wLength, Data) request with the following values:
    - wValue set to 15d (BOS)
    - windex set to 0
    - wLength set to wTotalLength (returned in previous call to GetDescriptor)
4. Test fails if the device does not return the descriptor or if the length is not wTotalLength. (9.1.1#1, 9.2#2)
5. Parse the data, keeping the BOS Power Delivery Capability Descriptor.
6. If bit 5 (CHARGING_POLICY) of USB Power Delivery Capability Descriptor is 0, end test with success.
7. Send a SetPDFeature request with the following values:
    a. wValue set to 54d (CHARGING_POLICY)
    b. wIndex to 0h
    c. Test fails if SetPDFeature fails (9.3.1#2)
8. Send a SetPDFeature request with the following values:
    a. wValue set to 54d (CHARGING_POLICY)
    b. wIndex set to 1h
    c. Test fails if the SetPDFeature fails (9.3.1#2)
9. Set a SetPDFeature request with the following values:
    a. wValue set to 54d (CHARGING_POLICY)
    b. wIndex set to 2h
    c. Test fails if the SetPDFeature fails (9.3.1#2)
10. Set a SetPDFeature request with the following values:
    a. wValue set to 54d (CHARGING_POLICY)
    b. wIndex set to 3h
    c. Test fails if the SetPDFeature fails (9.3.1#2)

9.9 USB Type-C Hub Bridge Test

Purpose	Verify that hub has Type-C Bridge
Critical for Safety	No

Applies to	USB Type-C hubs
Description	Make sure that USB Type-C hubs have a Bridge device.
Test setup	USB30CV
Preconditions	
Assertion References	9.1.1#2, 9.3#2
Device States Tested	Configured
Checklist References	

Test Procedure

1. Enumerate the DUT and put in required device state.
2. If DUT is not hub, end test with passing result.
3. Read Vendor Info Files (VIFs) for all Downstream Facing Ports.
4. If DUT is Enhanced Super Speed hub, find associated High Speed hub, and use this hub for remaining test steps.
5. If hub has at least one USB Type-C port (according to the VIFs), and does not have a USB Type-C Bridge device attached, then test fails (9.1.1#2, 9.3#2).
6. If hub does not have a USB Type-C port (according to the VIFs), and has a USB Type-C Bridge device, then test fails.