

USB PD CTS ENGINEERING CHANGE NOTIFICATION FORM

NOTICE: Any Company or Companies submitting a USB Power Delivery ECR proposal must be one of the following: a Promoter or Contributor of the USB 3.0 and 2.0 Specifications who have completed the USB Power Delivery addendum. If a group of Companies is submitting an ECR proposal, each company must be either a Promoter or Contributor of the USB 3.0 and 2.0 Specifications who have completed the USB Power Delivery addendum.

SPECIFICATION REVISIONS AND ADDENDA: At any point in time, there shall only be one current version of the USB PD CTS, termed the production version. At the same time, there may also be proposed revisions to the specification's design which are not yet approved and shall be held confidential as deemed necessary by the USB 3.0 and USB 2.0 Promoters and within the Group of Working Committee(s).

PROCEDURES FOR SUBMITTING PROPOSALS: Both members of the USB Implementers Forum as a whole and members of the USB 3.0 and USB 2.0 Promoters may submit requests to revise the USB PD CTS Specification. Such a request may be rejected or may result in a USB PD Engineering Change Notice (ECN), which is the official way USB specifications may be changed.

FORMAT OF PROPOSAL: The originator of a request to alter the USB PD CTS Specification may do so by posting this to the USB Power Delivery Compliance working group for review. Once the proposal has been reviewed by the working group it will be passed to the USB 3.0 and 2.0 Promoters for approval to publish.

RESUBMISSION AND APPEAL: The originator of a request that was not approved can redraft the original request. Rewritten proposal will be treated as a new proposal and will be evaluated using the procedures described above. The originator of a request that was not approved can also submit an appeal to the USB 3.0 and 2.0 Promoters. The appeal must be made in writing and addressed to the Secretary of the USB Implementers Forum.

ABOUT THE ENGINEERING CHANGE REQUEST FORM:

The Purpose of this Engineering Change Request Form is to expedite the review process of the proposal by providing explanations, background information, and examples of the proposed changes at a high level. This form serves as an executive summary to the actual proposal.

STEPS ON HOW TO SUBMIT A USB PD ENGINEERING CHANGE REQUEST:

- 1) Please fill out the Engineering Change Request Form on the following pages completely:
 - a) Detail the names and contact details for each of the ECR contributors
 - b) Update the ECR Title
 - c) Give a minimum of 2-3 sentences for each description on the form outlining the background to the ECR
- 2) For each section/table/figure to be updated:
 - a) Detail the section number, starting page and figure/table number to be updated as appropriate.
 - b) Detail existing text under "From Text"
 - c) Detail changed text under "To Text"
- 3) Save the file as "USB PD CTS 1.0 R 1" followed by the ECR Title as per step 1)b)
- 4) Post the ECR in the USB PD CTS Documents section under "ECR | New ECRs".
 - a) This ECR will then be reviewed by the Power Delivery Compliance Working Group.
 - b) Revisions to the ECR originating from the review should be submitted as document revision of the original ECR using "Add new document".

USB PD CTS ENGINEERING CHANGE NOTIFICATION FORM

Name: Yoon Lee Email: _____

Company: Apple Mailstop: _____

Address: _____

City: _____ State/Province: _____

Country: _____ Zip/Postal Code: _____

Phone: _____ FAX: _____

Name: _____ Email: _____

Company: _____ Mailstop: _____

Address: _____

City: _____ State/Province: _____

Country: _____ Zip/Postal Code: _____

Phone: _____ FAX: _____

Name: _____ Email: _____

Company: _____ Mailstop: _____

Address: _____

City: _____ State/Province: _____

Country: _____ Zip/Postal Code: _____

Phone: _____ FAX: _____

Name: _____ Email: _____

Company: _____ Mailstop: _____

Address: _____

City: _____ State/Province: _____

Country: _____ Zip/Postal Code: _____

Phone: _____ FAX: _____

USB PD CTS ENGINEERING CHANGE NOTIFICATION FORM

Title: tSenderResponse timeout test procedure update
Applied to: USB PD CTS Q1, 2026 OR

Brief description of the functional changes proposed:
--

Timing update per USB PD R3.2, V1.2 update.

Section 7.31.10.5 tSendHardReset

When a timer expiration leads to sending a Hard Reset, the transmission of the last bit of the Hard Reset Message Shall be completed within tSendHardReset. tSendHardReset applies to the expiration of:
--

- | |
|---|
| <ul style="list-style-type: none">• SenderResponseTimer,• SourcePPSCommTimer or• SourceEPRKeepAliveTimer. |
|---|

Table 7.9 Timer Values, tSenderResponse –“This timing can also be used when USBPD Device is in a PD2 explicit Contract”

Benefits as a result of the proposed changes:
--

Correct evaluation of the port behavior upon tSenderResponse timeout
--

An assessment of the impact to the existing revision and systems that currently conform to the USB specification:
--

None

An analysis of the hardware implications:
--

None

An analysis of the software implications:
--

None

An analysis of the compliance testing implications:
--

Tester to update the corresponding tests per the ECR updates.

USB PD CTS ENGINEERING CHANGE NOTIFICATION FORM

An analysis of the Vendor Info File (VIF) implications:

None

USB PD CTS ENGINEERING CHANGE NOTIFICATION FORM

Actual Change Requested

To Text:

TEST.PD.PROT.SRC.2 Get_Source_Cap No Request

Description: The Tester sends a *Get_Source_Cap* Message to the UUT. After receiving a *Source_Capabilities* Message, the Tester intentionally does not send the *Request* Message to force a *SenderResponse* Timer timeout on the Source UUT. The Tester verifies correct implementation of this timer.

Test Specific Tester Behavior: N/A

Test Conditions:

	Consumer Only	Provider Only, C/P, P/C, DRP
Rev2Src		✓
Rev3ChkdSrc		✓

Test Procedures:

- There are 2 possible bring-up procedures:
 - The UUT has VIF field PD_Port_Type set to Consumer/Provider, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7.
 - The UUT has VIF field PD_Port_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
- The Tester sends a *Get_Source_Cap* Message to the UUT.
- The check fails if the Tester does not receive *Source_Capabilities* Message. [TEST.PD.PROT.SRC.2#1]
- The Tester does not send a *Request* Message after receiving *Source_Capabilities* Message.
- The Tester checks that a *Hard Reset* is completed between *tSenderResponse* min and *tSenderResponse* max + *tSendHardReset* max, (see Table 1 Timing Table & Calculations)+ the delay is between the last bit of the *GoodCRC* Message EOP has been sent and the last bit of *Hard Reset* EOP has been received. [TEST.PD.PROT.SRC.2#2]

TEST.PD.PROT.SRC3.2 SenderResponseTimer Timeout

Description: As a Sink, the Tester intentionally does not send the *Request* Message to force a *SenderResponseTimer* timeout on the Source UUT. The Tester verifies correct implementation of this timer.

Test Specific Tester Behavior: N/A

Test Conditions:

	Consumer Only, C/P	Provider Only, DRP, P/C
Rev3ChkdSrc		✓

Test Procedures:

- The test starts in a disconnected state.
- The Tester applies Rd, and it waits for V_{BUS} for *tNoResponse* max.
- The check fails if the first bit of a *Source_Capabilities* Message is not received from the UUT within *tFirstSourceCap* max after V_{BUS} present. [TEST.PD.PROT.SRC3.2#1]
- The Tester intentionally does not send a *Request* Message and waits for a *Hard Reset*.
- The Tester checks that a *Hard Reset* is completed between *tSenderResponse* min and *tSenderResponse* max + *tSendHardReset* max. (see Table 1 Timing Table & Calculations), the delay is between the last bit of the *GoodCRC* Message EOP has been sent and the last bit of *Hard Reset* EOP has been received. [TEST.PD.PROT.SRC3.2#2]

USB PD CTS ENGINEERING CHANGE NOTIFICATION FORM

TEST.PD.PROT.SNK.6 SenderResponseTimer Timeout

Description: The Tester does not respond to the *Request* Message from the UUT, in order to force a SenderResponseTimer timeout on the UUT and verifies it is correctly implemented.

Test Specific Tester Behavior: N/A

Test Conditions:

	Consumer Only, P/C, DRP, C/P	Provider Only
<i>Rev2Snk</i>	✓	
<i>Rev3ChkdSnk</i>	✓	

Test Procedures:

- There are 2 possible bring-up procedures:
 - The UUT has VIF field PD_Port_Type set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8
 - The UUT has VIF field PD_Port_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
- The Tester transmits *Source Capabilities* Message with single.
 - B31...30 (Fixed Supply) set to 00b.
 - B29 (Dual-Role Power) set to 0b, Tester is a Source only.
 - B28 (USB Suspend Supported) set to 0b.
 - B26 (USB Communications Capable) set to 0b.
 - B25 (Dual-Role Data) set to 0b.
 - B24...22 (Reserved) set to 00b.
 - B21...20 (Peak Current) set to 00b.
 - B19...10 (Voltage) set to 5V.
 - B9...0 (Maximum Current) set to 500mA.
- The Tester continues to present *SinkTxNG* while waiting for a response if the test is running in PD3 mode. The check fails if a *Request* Message is not received from the UUT.
- The Tester does not send an *Accept* (as a response to the *Request* Message) in order to force a SenderResponseTimer timeout on the UUT.
- The Tester checks that a *Hard Reset* is completed between *tSenderResponse* min and *tSenderResponse* max + *tSendHardReset* max (see [Table 1](#) Timing Table & Calculations, the delay is between the last bit of the *GoodCRC* Message EOP has been sent and the last bit of *Hard Reset* EOP has been received. [TEST.PD.PROT.SNK.6#1])

TEST.PD.PROT.SNK.13 PR_Swap – Request SenderResponseTimer Timeout

Description: As a new Sink, the Tester intentionally does not send a *Request* Message after a *PR_Swap* in order to force a SenderResponseTimer timeout on the UUT and verifies it is correctly implemented.

Test Specific Tester Behavior:

- The Tester is a Dual-Role Power, initially a Source
- The Tester sends *Sink_Capabilities* setting B27 (Unconstrained Power) to 0b COMMON.PROC.PD.3

Test Conditions:

	Consumer Only, P/C, DRP, C/P	Provider Only
<i>Rev2Snk</i>	✓	

USB PD CTS ENGINEERING CHANGE NOTIFICATION FORM

Rev3ChkdSnk

✓

Test Procedures:

1. There are 2 possible bring-up procedures:
 - a. The UUT has VIF field PD_Port_Type set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8
 - b. The UUT has VIF field PD_Port_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
2. The Tester sends a *PR_Swap* Message. The Tester continues to present *SinkTxNG* while waiting for a response if the test is running in PD3 mode.
3. The Tester checks the response. [TEST.PD.PROT.SNK.13#1]
 - a. If the UUT responds with a *Wait* Message, the Tester resends *PR_Swap* every *tPRSwapWait* + **Error! Reference source not found.** for 10 times. The check fails and the test ends here if the Tester has resent *PR_Swap* for 10 times.
 - b. If the VIF field Accepts_PR_Swap_As_Snk = No, the check fails if the UUT responds with an *Accept* Message.
 - c. If the VIF field Accepts_PR_Swap_As_Snk = Yes, the check fails if the UUT responds with a *Reject* Message or Not Supported.
 - d. If the VIF field PD_Port_Type is set to “Consumer Only”, the Tester checks that the UUT responds with a *Reject* Message or *Not_Supported* Message. The test ends here.
4. At *tSrcTransition* max (this delay is from the last bit of *GoodCRC* sent in response to the *Accept* Message), the Tester drives the V_{BUS} voltage to *vSafe0V* within *tSrcSwapStdby* max.
5. The Tester presents Rd on the CC wire. The Tester sends a *PS_RDY* at the deadline limit of *tPSSourceOff* min, the delay is from the time the last bit of the EOP of the *GoodCRC* corresponding to the *Accept* Message.
6. The Tester checks that the UUT sends *PS_RDY* only after the UUT has applied *vSafe5V*. [TEST.PD.PROT.SNK.13#2]
7. The check fails if the *PS_RDY* from the UUT is not sent within *tPSSourceOn* min. [TEST.PD.PROT.SNK.13#3] This delay is from the *GoodCRC* sent by the UUT in response to the *PS_RDY* sent by the Tester.
8. The Tester checks that the UUT sends a *Source_Capabilities* Message after *tSwapSourceStart* min. [TEST.PD.PROT.SNK.13#4]
9. Once the Tester receives *Source_Capabilities* Message, the Tester establishes a PD contract using common procedure COMMON.PROC.PD.11.
10. The Tester waits **Error! Reference source not found.** to respond to messages from the UUT.
11. The Tester sends a *Get_Source_Cap* Message to the UUT.
12. The check fails if the Tester does not receive *Source_Capabilities* Message. [TEST.PD.PROT.SNK.13#5]
13. The Tester intentionally does not send a *Request* Message and waits for a *Hard Reset* Signaling.
14. The Tester checks that a *Hard Reset* Signaling is completed between *tSenderResponse* min and *tSenderResponse* max + *tSendHardReset* max (see Table 1 Timing Table & Calculations), the delay is between the last bit of the *GoodCRC* Message EOP has been sent and the last bit of *Hard Reset* EOP has been received. [TEST.PD.PROT.SNK.13#6]

USB PD CTS ENGINEERING CHANGE NOTIFICATION FORM

10 Appendix F: Timing table

Table 1 Timing Table & Calculations

Timer	Spec Rev.	min	max	Reference
<i>t_{SenderResponse}</i>	PD2	24 ms	30 ms*	[PD2 Spec]
	PD3	27 ms	50 ms	[PD3 Spec]

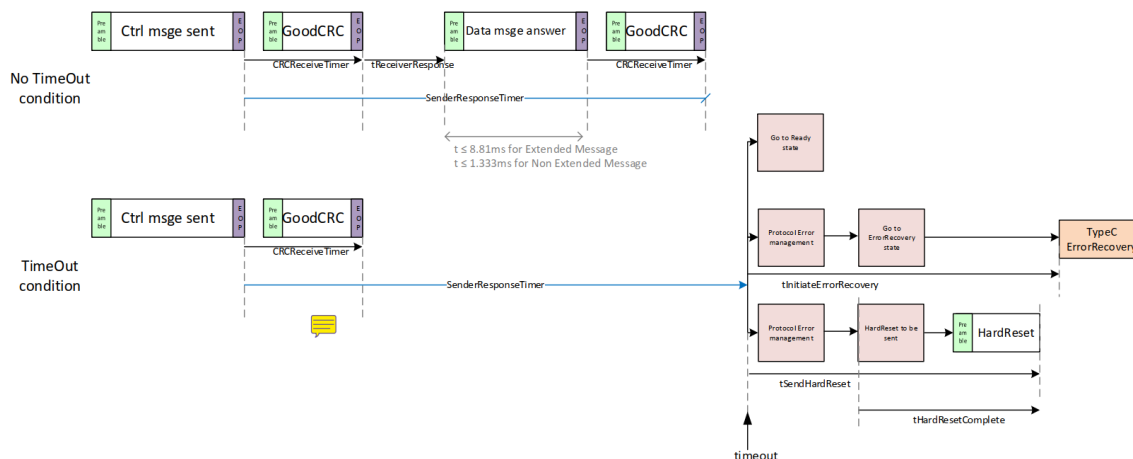
*The tester uses PD3 max timing on PD3 devices operating in PD2 mode.

USB PD CTS ENGINEERING CHANGE NOTIFICATION FORM

Background Info:

Illustration from USB PD R3.2 V1.1 ECN HardReset_processing_time_clarification:

Illustration of the clarification:



	What for ?	Min (ms)	Max (ms)
CRCReceiveTimer (tReceive)	Timeout	0.9	1.1
SenderResponseTimer(tSenderResponse)	Timeout	27	50
VDMResponseTimer (tVDMSenderResponse)	Timeout	24	50
tReceiverResponse	Max time to send response	-	15
tHardResetComplete	Max time to send message	4	5
tSendHardReset	Max time to send message		15
tInitiateErrorRecovery	Max time to ErrorRecovery		15

Note: Correction on SenderResponseTimer to start from the end of GoodCRC(discussed 3/25 in CTS call)