

USB Power Delivery ENGINEERING CHANGE NOTICE FORM

Title: Enter_USB

Applied to: USB Power Delivery Specification Revision 3.1
Version 1.2

Brief description of the functional changes proposed:
--

This ECN more clearly defines the usage of Reject and Wait in relation to Enter_USB and ensure consistency in the other usages and definitions.

Benefits as a result of the proposed changes:
--

Allows devices to either Reject an Enter_USB request or to postpone the request until the device is able to service it e.g. by re-negotiating its own power. Extends the definition and usage of Reject beyond that of invalid messages.
--

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

Hosts/devices will need to be able to handle cases where Wait is returned since this is not currently in the specification.

An analysis of the hardware implications:
--

None

An analysis of the software implications:
--

Some additional cases added by Wait in particular the option to retry.
--

An analysis of the compliance testing implications:
--

Additional compliance tests for Wait and note that it loosens the requirements for when Reject can be sent.

USB Power Delivery ENGINEERING CHANGE NOTICE FORM

Actual Change Requested

(a). Section 6.3.3: “Accept Message”

From Text:

The *Accept* Message is a *Valid* response in the following cases:

- It *Shall* be sent by the Source to signal the Sink that the Source is willing to meet the *Request* Message.
- It *Shall* be sent by the recipient of the *PR_Swap* Message to signal that it is willing to do a Power Role Swap and has begun the Power Role Swap sequence.
- It *Shall* be sent by the recipient of the *DR_Swap* Message to signal that it is willing to do a Data Role Swap and has begun the Data Role Swap sequence.
- It *Shall* be sent by the recipient of the *VCONN_Swap* Message to signal that it is willing to do a VCONN Swap and has begun the VCONN Swap sequence.
- It *Shall* be sent by the recipient of the *FR_Swap* Message to indicate that it has begun the Fast Role Swap sequence.
- It *Shall* be sent by the recipient of the *Soft_Reset* Message to indicate that it has completed its Soft Reset.

The *Accept* Message *Shall* be sent within *tReceiverResponse* of the receipt of the last bit of the Message (see Section 6.6.2).

To Text:

The *Accept* Message is a *Valid* response in the following cases:

- It *Shall* be sent by the Source to signal the Sink that the Source is willing to meet the *Request* Message.
- It *Shall* be sent by the Source to signal the Sink that the Source is willing to meet the *EPR_Request* Message.
- It *Shall* be sent by the recipient of the *PR_Swap* Message to signal that it is willing to do a Power Role Swap and has begun the Power Role Swap sequence.
- It *Shall* be sent by the recipient of the *DR_Swap* Message to signal that it is willing to do a Data Role Swap and has begun the Data Role Swap sequence.
- It *Shall* be sent by the recipient of the *VCONN_Swap* Message to signal that it is willing to do a VCONN Swap and has begun the VCONN Swap sequence.
- It *Shall* be sent by the recipient of the *FR_Swap* Message to indicate that it has begun the Fast Role Swap sequence.
- It *Shall* be sent by the recipient of the *Soft_Reset* Message to indicate that it has completed its Soft Reset.
- It *Shall* be sent by the recipient of the *Enter_USB* Message to indicate that it has begun the Enter USB sequence.

The *Accept* Message *Shall* be sent within *tReceiverResponse* of the receipt of the last bit of the Message (see Section 6.6.2).

(b). Section 6.3.4: “Reject Message”

From Text:

The *Reject* Message is a *Valid* response in the following cases:

USB Power Delivery ENGINEERING CHANGE NOTICE FORM

- It **Shall** be sent to signal the Sink that the Source is unable to meet the **Request** Message. This **May** be due an **Invalid** request or because the Source can no longer provide what it previously Advertised.
- It **Shall** be sent by the recipient of a **PR_Swap** Message to indicate it is unable to do a Power Role Swap.
- It **Shall** be sent by the recipient of a **DR_Swap** Message to indicate it is unable to do a Data Role Swap.
- It **Shall** be sent by the recipient of a **VCONN_Swap** Message that is not presently the VCONN Source, to indicate it is unable to do a VCONN Swap.

The **Reject** Message **Shall** be sent within **tReceiverResponse** of the receipt of the last bit of Message (see Section 6.6.2).

Note: the **Reject** Message is not a **Valid** response when a Message is not supported. In this case the **Not_Supported** Message is returned (see Section 6.3.16).

To Text:

The **Reject** Message is a **Valid** response in the following cases:

- It **Shall** be sent to signal the Sink, in **SPR Mode**, that the Source is unable to meet the **Request** Message. This **May** be due an **Invalid** request or because the Source can no longer provide what it previously Advertised.
- It **Shall** be sent to signal the Sink, in **EPR Mode**, that the Source is unable to meet the **EPR_Request** Message. This **May** be due an **Invalid** request or because the Source can no longer provide what it previously Advertised.
- It **Shall** be sent by the recipient of a **PR_Swap** Message to indicate it is unable to do a Power Role Swap.
- It **Shall** be sent by the recipient of a **DR_Swap** Message to indicate it is unable to do a Data Role Swap.
- It **Shall** be sent by the recipient of a **VCONN_Swap** Message that is not presently the VCONN Source, to indicate it is unable to do a VCONN Swap.
- It **Shall** be sent by UFP on receiving an **Enter_USB** Message to indicate it is unable to enter the requested USB Mode.

The sender of a **Request**, **EPR_Request**, **PR_Swap**, **DR_Swap**, **VCONN_Swap**, or **Enter_USB** Message, on receiving a **Reject** Message response, **Shall Not** send this same Message to the recipient again unless one of the following has occurred:

- A New Explicit Contract negotiation as a result of the Source sending a **Source_Capabilities** Message or **EPR_Source_Capabilities** Message. This can be triggered by:
 - The Source's Device Policy Manager.
 - A **Get_Source_Cap** Message sent from the Sink to the Source in **SPR Mode**.
 - An **EPR_Get_Source_Cap** Message sent from the Sink to the Source in **EPR Mode**.
 - A Power Role swap.
 - A Soft Reset.
 - A Hard Reset.
 - A Disconnect/Re-connect.
- A Data Role Swap.
- A Data Reset.

USB Power Delivery ENGINEERING CHANGE NOTICE FORM

The Sink **May** send a different **Request** Message to the one which was rejected but **Shall Not** repeat the same **Request** Message, using the same RDO, unless there has been a New Explicit Contract negotiation, Data Role Swap or Data Reset as described above.

The **Reject** Message **Shall** be sent within **tReceiverResponse** of the receipt of the last bit of Message (see Section 6.6.2).

Note: the **Reject** Message is not a **Valid** response when a Message is not supported. In this case the **Not_Supported** Message is returned (see Section 6.3.16).

(c). Section 6.3.12: “Wait Message”

From Text:

The **Wait** Message is a **Valid** response to a **Request**, a **PR_Swap**, **DR_Swap** or **VCONN_Swap** Message.

- It **Shall** be sent to signal the Sink that the Source is unable to meet the request at this time.
- It **Shall** be sent by the recipient of a **PR_Swap** Message to indicate it is unable to do a Power Role Swap at this time.
- It **Shall** be sent by the recipient of a **DR_Swap** Message to indicate it is unable to do a Data Role Swap at this time.
- It **Shall** be sent by the recipient of a **VCONN_Swap** Message that is not presently the VCONN Source to indicate it is unable to do a VCONN Swap at this time.

The **Wait** Message **Shall** be sent within **tReceiverResponse** of the receipt of the last bit of the Message (see Section 6.6.2).

To Text:

The **Wait** Message is a **Valid** response to one of the following Messages:

- It **Shall** be sent to signal the Sink, in response to a **Request** Message in SPR Mode during Power Negotiation, to indicate that the Source is **currently** unable to meet the request **at this time**.
- It **Shall** be sent to signal the Sink, in response to a **EPR_Request** Message in EPR Mode during Power Negotiation, to indicate that the Source is **currently** unable to meet the request **at this time**.
- It **Shall** be sent by the recipient of a **PR_Swap** Message to indicate it is **currently** unable to do a Power Role Swap **at this time**.
- It **Shall** be sent by the recipient of a **DR_Swap** Message to indicate it is **currently** unable to do a Data Role Swap **at this time**.
- It **Shall** be sent by the recipient of a **VCONN_Swap** Message that is not presently the VCONN Source to indicate it is **currently** unable to do a VCONN Swap **at this time**.
- It **Shall** be sent by the UFP on receipt of an **Enter_USB** Message to indicate it is **currently** unable to enter the requested USB Mode.

The **Wait** Message **Shall** be sent within **tReceiverResponse** of the receipt of the last bit of the Message (see Section 6.6.2).

(d). Section 6.3.12.5: “Wait in response to an Enter_USB Message”

New Text:

The **Wait** Message is used, by the UFP, when responding to an **Enter_USB** Message to indicate that entering the requested USB Mode might be possible in the future. This can occur, for example, in any case where the UFP needs to negotiate more power in order to enter the mode. Once the UFP has completed this the DFP **Should** initiate the Enter USB process again by sending an **Enter_USB** Message.

A DFP that receives a **Wait** Message in response to an **Enter_USB** Message **Shall** wait **tEnterUSBWait** after receiving the **Wait** Message before sending another **Enter_USB** Message.

(e). Section 6.4.8: “Enter_USB Message”

USB Power Delivery ENGINEERING CHANGE NOTICE FORM

From Text:

The **Enter_USB** Message **Shall** be sent by the DFP to its UFP Port Partner and to the Cable Plug(s) of an Active Cable, when in an Explicit Contract, to enter a specified USB Mode of operation. The recipient of the Message **Shall** respond by sending either an **Accept** Message in response to a **Valid** request Or a **Reject** Message in response to an **Invalid** request.

To Text:

The **Enter_USB** Message **Shall** be sent by the DFP to its UFP Port Partner and to the Cable Plug(s) of an Active Cable, when in an Explicit Contract, to enter a specified USB Mode of operation. The recipient of the Message **Shall** respond by sending an **Accept** Message, a **Wait** Message or a **Reject** Message (see Section 6.9).

Commented [RPM1]: Look at cross-references elsewhere.

(f). Section 6.6.4.5: “tEnterUSBWait”

New Text:

The time before the next **Enter_USB** Message, after a **Wait** Message has been received in response to a **Enter_USB** Message is a minimum of **tEnterUSBWait** min (see Section 6.3.12). The DFP **Shall** wait at least **tEnterUSBWait** after receiving the **EOP** of a **Wait** Message sent in response to an **Enter_USB** Message, before sending a new **Enter_USB** Message.

(g). Table 6-68 “Time Values”

New Text:

Parameter	Value (min)	Value (nom)	Value (max)	Unit s	Reference
tEnterUSBWait	100			ms	Section 6.6.4.5

Commented [RPM2]: Will take a look at collapsing timers editorially into a single timer.

(h). Section 6.9 “Accept, Reject and Wait”

New Text:

The recipient of a **Request**, **EPR_Request**, **PR_Swap**, **DR_Swap**, **VCONN_Swap**, or **Enter_USB** Message **Shall** respond by sending one of the following responses:

- an **Accept** Message in response to a **Valid** request which can be serviced immediately (see Section 6.3.3).
- a **Wait** Message in response to a **Valid** request which cannot be serviced immediately but could be serviced at a later time (see Section 6.3.12).
- a **Reject** Message in response to an **Invalid** request or a request which is outside of the device's design capabilities (see Section 6.3.4).

(i). Section 8.3.3.15 “Enter_USB Message State Diagrams”

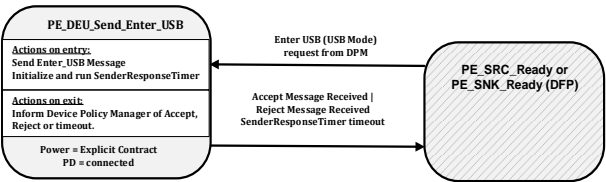
From Text:

8.3.3.15.1 DFP Enter_USB Message State Diagrams

Figure 8-115 shows the state diagram for an **Enter_USB** Message sent by a DFP.

USB Power Delivery ENGINEERING CHANGE NOTICE FORM

Figure 8-115 DFP Enter_USB Message State Diagram



8.3.3.15.1 PE_DEU_Send_Enter_USB State

The **PE_DEU_Send_Enter_USB** State *Shall* be entered from the **PE_SRC_Ready** or **PE_SNK_Ready** State when requested by the Device Policy Manager and the Port is operating as a DFP.

On entry to the **PE_DEU_Send_Enter_USB** State the Policy Engine *Shall* request the Protocol Layer to send an **Enter_USB** Message and then initialize and run the **SenderResponseTimer**.

On exit from the **PE_DEU_Send_Enter_USB** state the Policy Engine *Shall* inform the Device Policy Manager of the outcome: **Accept** Message received, **Reject** Message received, **SenderResponseTimer** timeout.

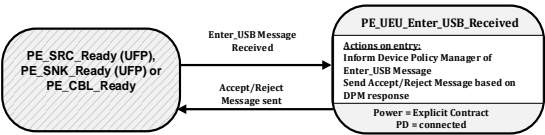
The Policy Engine *Shall* transition back to the **PE_SRC_Ready** or **PE_SNK_Ready** State depending on the Ports power role when:

- An **Accept** Message has been received or
- A **Reject** Message has been received
- There is a **SenderResponseTimer** timeout.

8.3.3.15.2 UFP or Cable Plug Enter_USB Message State Diagrams

Figure 8-116 shows the state diagram for an **Enter_USB** Message received by a UFP or Cable Plug.

Figure 8-116 UFP Enter_USB Message State Diagram



8.3.3.15.2.1 PE_UEU_Enter_USB_Received State

The **PE_UEU_Enter_USB_Received** state *Shall* be entered from the **PE_SRC_Ready**, **PE_SNK_Ready** or **PE_CBL_Ready** state as appropriate (see Figure 8-84, Figure 8-85 and Figure 8-148) when an **Enter_USB** Message is received and the Port is operating as a UFP or is a Cable Plug.

On entry to the **PE_UEU_Enter_USB_Received** state the Policy Engine *Shall* inform the Device Policy Manager. The Device Policy Manager responds with an indication of whether the **Enter_USB** Message is to be accepted or rejected. The Policy Engine *Shall* send either an **Accept** Message or a **Reject** Message as appropriate.

The Policy Engine *Shall* transition back to the **PE_SRC_Ready**, **PE_SNK_Ready** or **PE_CBL_Ready** state as appropriate when:

- Either an **Accept** Message or a **Reject** Message has been sent.
-

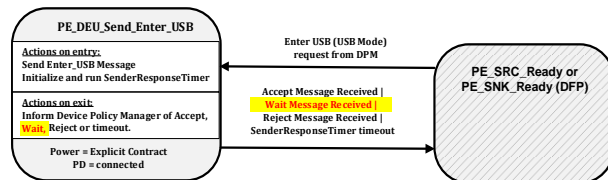
To Text:

8.3.3.15.1 DFP Enter_USB Message State Diagrams

Figure 8-115 shows the state diagram for an **Enter_USB** Message sent by a DFP.

USB Power Delivery ENGINEERING CHANGE NOTICE FORM

Figure 8-115 DFP Enter_USB Message State Diagram



8.3.3.15.1.1

PE_DEU_Send_Enter_USB State

The **PE_DEU_Send_Enter_USB** State *Shall* be entered from the **PE_SRC_Ready** or **PE_SNK_Ready** State when requested by the Device Policy Manager and the Port is operating as a DFP.

On entry to the **PE_DEU_Send_Enter_USB** State the Policy Engine *Shall* request the Protocol Layer to send an **Enter_USB** Message and then initialize and run the **SenderResponseTimer**.

On exit from the **PE_DEU_Send_Enter_USB** state the Policy Engine *Shall* inform the Device Policy Manager of the outcome: **Accept** Message received, **Reject** Message received, **SenderResponseTimer** timeout.

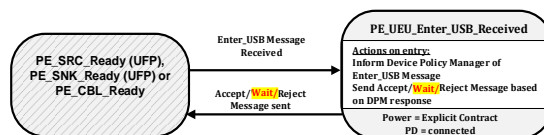
The Policy Engine *Shall* transition back to the **PE_SRC_Ready** or **PE_SNK_Ready** State depending on the Ports power role when:

- An **Accept** Message has been received or
- A **Wait Message has been received** or
- A **Reject** Message has been received **or**
- There is a **SenderResponseTimer** timeout.

8.3.3.15.2 UFP or Cable Plug Enter_USB Message State Diagrams

Figure 8-116 shows the state diagram for an **Enter_USB** Message received by a UFP or Cable Plug.

Figure 8-116 UFP Enter_USB Message State Diagram



8.3.3.15.2.1

PE_UEU_Enter_USB_Received State

The **PE_UEU_Enter_USB_Received** state *Shall* be entered from the **PE_SRC_Ready**, **PE_SNK_Ready** or **PE_CBL_Ready** state as appropriate (see Figure 8-84, Figure 8-85 and Figure 8-148) when an **Enter_USB** Message is received and the Port is operating as a UFP or is a Cable Plug.

On entry to the **PE_UEU_Enter_USB_Received** state the Policy Engine *Shall* inform the Device Policy Manager. The Device Policy Manager responds with an indication of whether the **Enter_USB** Message is to be accepted or rejected. The Policy Engine *Shall* send either an **Accept** Message, a **Wait Message** or a **Reject** Message as appropriate.

The Policy Engine *Shall* transition back to the **PE_SRC_Ready**, **PE_SNK_Ready** or **PE_CBL_Ready** state as appropriate when:

- Either an **Accept** Message, a **Wait Message** or a **Reject** Message has been sent.