

USB Type-C® ENGINEERING CHANGE NOTICE FORM

Title: Clarify ErrorRecovery State

Applied to: USB Type-C® Specification Release 2.0, August 2019

Brief description of the functional changes proposed:
--

Clarifies that ports that support Power Delivery are also required to support the ErrorRecovery state.
--

Benefits as a result of the proposed changes:
--

Makes it clear that when PD Hard Reset fails the only logical step that can be taken to ‘fix’ the problem is to have a clean start-over by simulation of an unplug followed by a plug in.

Implementations that go to the Detached or Unattached states have no chance of recovery and the transition into those states are NOT detectable during compliance testing.
--

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

Devices that do not support the ErrorRecovery state will continue to work as they do today.

An analysis of the hardware implications:
--

A Sink that is driving ErrorRecovery may see up to 5V on CC which may exceed their process transistor breakdown voltage. There is another ECR in flight that will allow Sinks to clamp CC at some voltage around 3V to avoid this issue.
--

An analysis of the software implications:
--

None.

An analysis of the compliance testing implications:
--

Simplifies compliance testing as there is only one behavior when testing ports that support PD.

USB Type-C® ENGINEERING CHANGE NOTICE FORM

Actual Change Requested

(a). Section 4.5.2.2.2

From Text:

4.5.2.2.2 ErrorRecovery State

This state appears in Figure 4-12, Figure4-13, Figure 4-14, Figure 4-16, and Figure 4-17.

The [ErrorRecovery](#) state is where the port removes the terminations from the CC1 and CC2 pins for [tErrorRecovery](#) followed by transitioning to the appropriate [Unattached.SNK](#) or [Unattached.SRC](#) state based on port type. This is the equivalent of forcing a detach event and looking for a new attach.

The port should transition to the [ErrorRecovery](#) state from any other state when directed.

A port may choose not to support the [ErrorRecovery](#) state. If the [ErrorRecovery](#) state is not supported, the port shall be directed to the [Disabled](#) state if supported. If the [Disabled](#) state is not supported, the port shall be directed to either the [Unattached.SNK](#) or [Unattached.SRC](#) states.

To Text:

4.5.2.2.2 ErrorRecovery State

This state appears in Figure 4-12, Figure4-13, Figure 4-14, Figure 4-16, and Figure 4-17.

The [ErrorRecovery](#) state is where the port removes the terminations from the CC1 and CC2 pins for [tErrorRecovery](#) followed by transitioning to the appropriate [Unattached.SNK](#) or [Unattached.SRC](#) state based on port type. This is the equivalent of forcing a detach event and looking for a new attach.

Ports that support USB Power Delivery shall support the [ErrorRecovery](#) state.

Ports that support the [ErrorRecovery](#) state shall transition to the [ErrorRecovery](#) state from any other state when directed.

A port [that does not support USB Power Delivery may](#) choose not to support the [ErrorRecovery](#) state. If the [ErrorRecovery](#) state is not supported, the port shall be directed to the [Disabled](#) state if supported. If the [Disabled](#) state is not supported, the port shall be directed to either the [Unattached.SNK](#) or [Unattached.SRC](#) states.