Evaluation Copy

Compute Express Link™ (CXL™)

Engineering Change Notice to the Specification 2.0

September 2021

Component State Dump Log

LEGAL NOTICE FOR THIS PUBLICLY-AVAILABLE SPECIFICATION FROM COMPUTE EXPRESS LINK CONSORTIUM, INC.

© 2019-2021 COMPUTE EXPRESS LINK CONSORTIUM, INC. ALL RIGHTS RESERVED.

This CXL **Specification** (this "**CXL Specification**" or this "**document**") is owned by and is proprietary to Compute Express Link Consortium, Inc., a Delaware nonprofit corporation (sometimes referred to as "**CXL**" or the "**CXL Consortium**" or the "**Company**") and/or its successors and assigns.

NOTICE TO USERS WHO ARE MEMBERS OF THE CXL CONSORTIUM:

If you are a Member of the CXL Consortium (sometimes referred to as a "**CXL Member**"), and even if you have received this publicly-available version of this CXL Specification after agreeing to CXL Consortium's Evaluation Copy Agreement (a copy of which is available https://www.computeexpresslink.org/download-the-specification, each such CXL Member must also be in compliance with all of the following CXL Consortium documents, policies and/or procedures (collectively, the "**CXL Governing Documents**") in order for such CXL Member's use and/or implementation of this CXL Specification to receive and enjoy all of the rights, benefits, privileges and protections of CXL Consortium membership: (i) CXL Consortium's Intellectual Property Policy; (ii) CXL Consortium's Bylaws; (iii) any and all other CXL Consortium policies and procedures; and (iv) the CXL Member's Participation Agreement.

NOTICE TO NON-MEMBERS OF THE CXL CONSORTIUM:

If you are **not** a CXL Member and have received this publicly-available version of this CXL Specification, your use of this document is subject to your compliance with, and is limited by, all of the terms and conditions of the CXL Consortium's Evaluation Copy Agreement (a copy of which is available at https://www.computeexpresslink.org/download-the-specification**).**

In addition to the restrictions set forth in the CXL Consortium's Evaluation Copy Agreement, any references or citations to this document must acknowledge the Compute Express Link Consortium, Inc.'s sole and exclusive copyright ownership of this CXL Specification. The proper copyright citation or reference is as follows: **"© 2019-2021 COMPUTE EXPRESS LINK CONSORTIUM, INC. ALL RIGHTS RESERVED."** When making any such citation or reference to this document you are not permitted to revise, alter, modify, make any derivatives of, or otherwise amend the referenced portion of this document in any way without the prior express written permission of the Compute Express Link Consortium, Inc.

Except for the limited rights explicitly given to a non-CXL Member pursuant to the explicit provisions of the CXL Consortium's Evaluation Copy Agreement which governs the publicly-available version of this CXL Specification, nothing contained in this CXL Specification shall be deemed as granting (either expressly or impliedly) to any party that is not a CXL Member: (ii) any kind of license to implement or use this CXL Specification or any portion or content described or contained therein, or any kind of license in or to any other intellectual property owned or controlled by the CXL Consortium, including without limitation any trademarks of the CXL Consortium.; or (ii) any benefits and/or rights as a CXL Member under any CXL Governing Documents.

LEGAL DISCLAIMERS FOR ALL PARTIES:

THIS DOCUMENT AND ALL SPECIFICATIONS AND/OR OTHER CONTENT PROVIDED HEREIN IS PROVIDED ON AN "**AS IS**" BASIS. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, COMPUTE EXPRESS LINK CONSORTIUM, INC. (ALONG WITH THE CONTRIBUTORS TO THIS DOCUMENT) HEREBY DISCLAIM ALL REPRESENTATIONS, WARRANTIES AND/OR COVENANTS, EITHER EXPRESS OR IMPLIED, STATUTORY OR AT COMMON LAW, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, VALIDITY, AND/OR NON-INFRINGEMENT.
In the event this CXL Specification makes any references (including without limitation any incorporation by reference) to another standard's setting organization's or any other party's ("**Third Party**") content or work, including without limitation any specifications or standards of any such Third Party ("**Third Party Specification**"), you are hereby notified that your use or implementation of any Third Party Specification: (i) is not governed by any of the CXL Governing Documents; (ii) may require your use of a Third Party's patents, copyrights or other intellectual property rights, which in turn may require you to independently obtain a license or other consent from that Third Party in order to have full rights to implement or use that Third Party Specification; and/or (iii) may be governed by the intellectual property policy or other policies or procedures of the Third Party which owns the Third Party Specification. Any trademarks or service marks of any Third Party which may be referenced in this CXL Specification is owned by the respective owner of such marks.

NOTICE TO ALL PARTIES REGARDING THE PCI-SIG UNIQUE VALUE PROVIDED IN THIS CXL SPECIFICATION:

NOTICE TO USERS: THE UNIQUE VALUE THAT IS PROVIDED IN THIS CXL SPECIFICATION IS FOR USE IN VENDOR DEFINED MESSAGE FIELDS, DESIGNATED VENDOR SPECIFIC EXTENDED CAPABILITIES, AND ALTERNATE PROTOCOL NEGOTIATION ONLY AND MAY NOT BE USED IN ANY OTHER MANNER, AND A USER OF THE UNIQUE VALUE MAY NOT USE THE UNIQUE VALUE IN A MANNER THAT (A) ALTERS, MODIFIES, HARMS OR DAMAGES THE TECHNICAL FUNCTIONING, SAFETY OR SECURITY OF THE PCI-SIG ECOSYSTEM OR ANY PORTION THEREOF, OR (B) COULD OR WOULD REASONABLY BE DETERMINED TO ALTER, MODIFY, HARM OR DAMAGE THE TECHNICAL FUNCTIONING, SAFETY OR SECURITY OF THE PCI-SIG ECOSYSTEM OR ANY PORTION THEREOF (FOR PURPOSES OF THIS NOTICE, "**PCI-SIG ECOSYSTEM**" MEANS THE PCI-SIG SPECIFICATIONS, MEMBERS OF PCI-SIG AND THEIR ASSOCIATED PRODUCTS AND SERVICES THAT INCORPORATE ALL OR A PORTION OF A PCI-SIG SPECIFICATION AND EXTENDS TO THOSE PRODUCTS AND SERVICES INTERFACING WITH PCI-SIG MEMBER PRODUCTS AND SERVICES).

Early Copy

Evaluation Copy

CXL ENGINEERING CHANGE NOTICE

TITLE:	Component State Dump Log
DATE:	Introduced: 03/09/2021 Latest Version: 05/18/2021
AFFECTED DOCUMENT:	CXL 2.0
SPONSOR:	Ariel Sibley, Microchip Chris Petersen, Facebook

Part I

1. Summary of Functional Changes

This ECN defines a new Component State Dump Log for reading state dump information from a component. The Component State Dump Log supports being populated automatically under vendor specific conditions (e.g. crash, severe error), and manually populated using a newly introduced Populate Log command.

This ECN also introduces the concept of Clear Log, Populate Log, and Get Log Capabilities.

Clear Log was introduced to provide a mechanism to clear the Component State Dump Log. This allows a mode of operation where the oldest Component State Dump Log is preserved until it is cleared using Clear Log.

Populate Log was introduced to provide a mechanism to populate the Component State Dump Log on demand.

Get Log Capabilities was introduced to provide a mechanism to determine if a given Log Identifier UUID supports Clear Log, and/or Populate Log, and/or Auto Populate, and/or is persistent across cold reset.

2. Benefits as a Result of the Changes

When a device's control path encounters a fatal error and can no longer continue normal operation, this is known as a "crash". In order to debug crashes, it is often useful to capture the component's state at the time of the crash. It may also be useful to capture a component's state on demand when debugging other types of issues. These state dumps and run-time logging are orthogonal use cases. If Vendor Log is used to report both state dump and run-time logs, the host will be forced to dump the previous state dump blob each time the current run-time logs are requested. In components where state dump is maintained across cold reset, this may become a persistent condition in the device.

A vendor defined UUID could be used to implement a state dump log, but since state dump is anticipated to be useful for most components, it makes sense to have a designated UUID. The current specification also does not provide a mechanism to clear

Evaluation Copy

logs, so the addition of Clear Log avoids the need for a vendor defined command to perform this operation. Finally, the current specification does not provide a mechanism to populate a log on demand, so the addition of Populate Log avoids the need for a vendor defined command to perform this operation.

This ECN allows state dump information to be extracted using a spec defined mechanism, which enables industry standard host tooling to collect this information.

3. **Assessment of the Impact**
   Components may optionally implement support for the new Log Identifier UUID. Components may optionally also implement support for Clear Log, Populate Log, and Get Log Capabilities.

   Host software may optionally implement support for the new Log Identifier UUID, Get Log Capabilities, Clear Log, and Populate Log.

4. **Analysis of the Hardware Implications**
   Components implementing Log Identifiers in hardware may optionally implement support for the new Log Identifier UUID. Components implementing opcodes in hardware may optionally implement support for Clear Log, Populate Log, and Get Log Capabilities.

5. **Analysis of the Software Implications**
   Host software may optionally implement support for the new Log Identifier UUID, Get Log Capabilities, Clear Log, and Populate Log.

6. **Analysis of the Compliance and Test Implications**
   N/A

Part II

Detailed Description of the change

Section 8.2.8.4.5.1 Command Return Codes:

- Table 150: **Added** Interrupted (0018h) return code.

Section 8.2.9 CXL Device Command Interface:

- Table 152: **Added** three new opcodes:
  - 0402h: Get Log Capabilities
  - 0403h: Clear Log
  - 0404h: Populate Log
- Section 8.2.9.4 Logs
  - Section 8.2.9.4.1 Get Supported Logs (Opcode 0400h)
    - Table 169: **Added** Component State Dump Log UUID
  - Section 8.2.9.4.2 Get Log (Opcode 0401h)
    - Table 170: **Added** Component State Dump Log UUID
    - **Added** Section 8.2.9.4.2.a: Component State Dump Log
  - **Added** Section 8.2.9.4.a Get Log Capabilities (Opcode 0402h)

- **Added** Section 8.2.9.4.b Clear Log (Opcode 0403h)
- **Added** Section 8.2.9.4.c Populate Log (Opcode 0404h)

Evaluation Copy

8.2.8.4.5.1 Command Return Codes

Table 150. Command Return Codes

Value	Definition
0018h	**Interrupted:** The command could not be completed successfully due to an asynchronous event.

8.2.9.4.1 Get Supported Logs (Opcode 0400h)

Retrieve the list of device specific logs (identified by UUID) and the maximum size of each Log.

Possible Command Return Codes:

- Success
- Internal Error
- Retry Required
- Invalid Payload Length

Command Effects:

- None

Table 168. Get Supported Logs Output Payload

Byte Offset	Length	Description
0	2	**Number of Supported Log Entries:** The number of Supported Log Entries returned in the output payload.
2	6	**Reserved**
8	Varies	**Supported Log Entries:** Device specific list of supported log identifier UUIDs and the current size of each log

Table 169. Get Supported Logs Supported Log Entry

Byte Offset	Length	Description
0	10h	**Log Identifier:** UUID representing the log to retrieve data for. The following Log Identifier UUIDs are defined in this specification: • 0da9c0b5-bf41-4b78-8f79-96b1623b3f17 – Command Effects Log (CEL) • 5e1819d9-11a9-400c-811f-d60719403d86 – Vendor Debug Log • b3fab4cf-01b6-4332-943e-5e9962f23567 – Component State Dump Log
10h	4	**Log Size:** The number of bytes of log data available to retrieve for the log identifier.

8.2.9.4.2 Get Log (Opcode 0401h)

Retrieve a log from the device, identified by a specific UUID. The host shall retrieve the size of the log first using the Get Supported Logs command, then issue enough of these commands to retrieve all the log information, incrementing the Log Offset each time. The device shall return Invalid Parameter if the Offset or Length fields attempt to access beyond the size of the log as reported by Get Supported Logs.

Possible Command Return Codes:

- Success
- Invalid Parameter
- Internal Error

- Retry Required
- Invalid Payload Length
- Interrupted

Command Effects:

- None

Table 170. Get Log Input Payload

Byte Offset	Length	Description
0	10h	**Log Identifier:** UUID representing the log to retrieve data for. The following Log Identifier UUIDs are defined in this specification: • 0da9c0b5-bf41-4b78-8f79-96b1623b3f17 – Command Effects Log (CEL) • 5e1819d9-11a9-400c-811f-d60719403d86 – Vendor Debug Log • b3fab4cf-01b6-4332-943e-5e9962f23567 – Component State Dump Log
10h	4	**Offset:** The byte offset in the log data to return in the output payload.
14h	4	**Length:** Length in bytes of log data to return in the output payload.

8.2.9.4.2.a Component State Dump Log

Log Identifier: b3fab4cf-01b6-4332-943e-5e9962f23567

The Component State Dump Log is an optional method for allowing vendor specific state information to be extracted using standard drivers.

The Component State Dump Log can be populated in two ways:

- Auto populate.
- Manual populate using Populate Log.

A component that supports the Component State Dump Log shall support at least one of the above methods.

The two methods and their associated trigger requirements are detailed in Table A. The Component State Dump Log shall be populated by a given method if the trigger occurs, and the logical AND of all the conditions for that trigger is true.

Table A. Component State Dump Log Population Methods and Triggers

Method	Trigger	Condition	Condition Reference
Auto Populate	Vendor Specific	Auto Populate Trigger Count Since Clear = 0	Table B. Component State Dump Log Format
		Auto Populate Supported = 1	Table D. Get Log Capabilities Output Payload
Manual Populate	Populate Log command received	Log Identifier = b3fab4cf-01b6-4332-943e-5e9962f23567	Table F. Populate Log Input Payload
		Populate Log Supported = 1	Table D. Get Log Capabilities Output Payload

The trigger for the auto populate method is vendor specific, but one example may be a severe internal error in the component.

When a population method triggers and all required conditions are met, any existing Component State Dump Data is cleared before populating the new log contents.

The log contents should persist across cold reset. The component shall indicate whether the log persists across cold reset using the Persistent Across Cold Reset bit in the Get Log Capabilities Output Payload.

Evaluation Copy

If the component has Component State Dump Data available to be reported in the Component State Dump Log after a subsequent reset, the Component State Dump Log contents shall be available when the Mailbox Interfaces Ready bit in the Memory Device Status Register is set to 1.

To handle corner cases related to an existing Component State Dump Log being overwritten by an Auto Populate trigger while host software is reading the existing contents of the log, host software must begin each Component State Dump Log fetch sequence by issuing a Get Log command with Offset = 0, followed by zero or more Get Log commands with non-zero offset. If the component is reset, host software must start a new fetch sequence.

If a Get Log command with non-zero Offset is received requesting the Component State Dump Log, the component shall apply the first applicable case from the list below:

- Return Invalid Input if the component has not previously returned Success for a Get Log command with Offset = 0 requesting the Component State Dump Log.
- Return Interrupted if the contents of the Component State Dump Log have changed since the last time the component returned Success for a Get Log command with Offset = 0 requesting the Component State Dump Log.
- Return Success and provide the log contents of the specified offset corresponding to the state of the Component State Dump Log when the current fetch sequence began (i.e. when the last Get Log command with Offset = 0 requesting the Component State Dump Log was completed with a return code of Success).

If a Get Log command with Offset = 0 is received requesting the Component State Dump Log, the component shall apply the first applicable case from the list below:

- ~~Return Interrupted if the Component State Dump Log was manually populated using Populate Log, and then subsequently overwritten by an Auto Populate operation prior to returning Success for a Get Log command with Offset = 0 requesting the Component State Dump Log.~~
- Return Retry Required if the log contents are not fully populated.
- Return Success and provide the log contents starting at Offset = 0.

When starting a new fetch sequence for a Component State Dump Log that was manually populated using Populate Log, host software should issue a single Get Log command requesting the Component State Dump Log with Offset = 0 and Length >= 24h. Host software should verify that the Auto Populate Data flag is set to 0. If the Auto Populate Data flag is set to 1, this indicates the manually populated Component State Dump Log was overwritten by an Auto Populate operation, and the Component State Dump Data corresponds to the results of that Auto Populate operation.

If an Auto Populate trigger occurs while the component is processing a Manual Populate operation triggered by a Populate Log command, and the component is still capable of returning a completion, the component shall complete the Populate Log command with a return code of Interrupted.

The format of the Component State Dump Log is defined in Table B.

Table B. Component State Dump Log Format

Byte Offset	Length	Description
00h	04h	**Component State Dump Data Length:** Length of the Component State Dump Data field in bytes.
04h	01h	**Auto Populate Trigger Count Since Clear:** The number of times the component has encountered the Trigger for the Auto Populate method since the last time the Component State Dump Log was cleared. Tracking this value is optional. If this value is tracked, it should persist across cold reset. This value is cleared to 0 when the Component State Dump Log is cleared using Clear Log, or the Component State Dump Log is populated using Populate Log.

Evaluation Copy

		If the component tracks this value, and the Auto Populate Data bit in the Flags field is set to 1, this field shall be at least 1. If the component does not track this value, this field shall be set to 0. This value shall saturate at FFh instead of rolling over to 0.
05h	01h	**Event Log:** The Event Log, as defined in Table 158 (Get Event Records Input Payload), containing the Associated Event Record Handle. If the Associated Event Record Handle is 0, the value of this field is undefined.
06h	02h	**Associated Event Record Handle:** The Event Record Handle, as defined in Table 153 (Common Event Record Format), corresponding to an Event Record that is associated with the Auto Populate trigger that generated the Component State Dump Data. If there are no associated Event Records, this field shall be set to 0. If there was an associated Event Record, but it is no longer in the Event Log, this field shall be set to 0. If the Auto Populate Data bit in the Flags field is set to 0, this field is undefined.
08h	08h	**Timestamp:** The Timestamp, as defined by the Timestamp field in Table 166 (Get Timestamp Output Payload), at the time the Component State Dump Data was generated.
10h	10h	**Component State Dump Format UUID:** Optional value to uniquely identify the format of the Component State Dump Data field. A value of 0 indicates that the format of the Component State Dump Data is not indicated.
20h	04h	**Flags** Bit[0]: Auto Populate Data. Set to 1 if the Component State Dump Data was generated using the Auto Populate method. Set to 0 if the Component State Dump Data Length is 0, or the Component State Dump Data was generated using Populate Log. Bit[31:1]: Reserved.
24h	1Ch	**Reserved**
40h	Varies	**Component State Dump Data:** Vendor specific.

Table 152. CXL Device Command Opcodes

Command Set Bits[15:8]		Command		Combined Opcode	Required*	Input Payload Size (B)	Output Payload Size (B)
04h	Logs	00h	Get Supported Logs (Section 8.2.9.4.1)	0400h	M	0	8+
		01h	Get Log (Section 8.2.9.4.2)	0401h	M	18h	0+
		02h	Get Log Capabilities (Section 8.2.9.4.a)	0402h	O	10h	4
		03h	Clear Log (Section 8.2.9.4.b)	0403h	O	10h	0
		04h	Populate Log (Section 8.2.9.4.c)	0404h	O	10h	0

8.2.9.4.a Get Log Capabilities (Opcode 0402h)

Gets capabilities related to the specified log. If the component supports this command, it shall be implemented for all Log Identifier UUIDs that the component supports. This command shall return Invalid Log if the specified Log Identifier is not supported by the component.

Possible Command Return Codes:

- Success
- Invalid Log
- Internal Error
- Unsupported

Evaluation Copy

Command Effects:

- None

Table C. Get Log Capabilities Input Payload

Byte Offset	Length	Description
00h	10h	**Log Identifier:** UUID representing the log to get capabilities for.

Table D. Get Log Capabilities Output Payload

Byte Offset	Length	Description
00h	04h	**Parameter Flags:** Bit[0]: Clear Log Supported. This bit is set to 1 if the log supports being cleared using the Clear Log command. Bit[1]: Populate Log Supported. This bit is set to 1 if the log supports being populated using the Populate Log command. Bit[2]: Auto Populate Supported. This bit is set to 1 if the log supports being auto populated. Details on the meaning of this bit are specific to each Log Identifier UUID. Bit[3]: Persistent Across Cold Reset. This bit is set to 1 if the log is persistent across cold reset. Bit[31:4]: Reserved.

8.2.9.4.b Clear Log (Opcode 0403h)

Clears the contents of the specified log.

This command shall return Invalid Log if the specified Log Identifier is not supported by the component.

This command shall return Invalid Input if the specified Log Identifier does not have the Clear Log Supported bit set to 1 in the Get Log Capabilities Output Payload.

Possible Command Return Codes:

- Success
- Invalid Input
- Invalid Log
- Internal Error
- Unsupported

Command Effects:

- Immediate Log Change

Table E. Clear Log Input Payload

Byte Offset	Length	Description
00h	10h	**Log Identifier:** UUID representing the log to clear.

8.2.9.4.c Populate Log (Opcode 0404h)

Populates the contents of the specified log.

This may be a background operation. If the component implements this command as a background operation for any supported Log Identifier, the Background Operation bit in the Command Effects Log entry for Populate Log shall be set to 1.

This command shall return Invalid Log if the specified Log Identifier is not supported by the component.

Evaluation Copy

This command shall return Invalid Input if the specified Log Identifier does not have the Populate Log Supported bit set to 1 in the Get Log Capabilities Output Payload.

Possible Command Return Codes:

- Success
- Background Command Started
- Invalid Input
- Invalid Log
- Internal Error
- Unsupported
- Interrupted

Command Effects:

- Immediate Log Change
- Background Operation (if the component implements this command as a background operation for any supported Log Identifier)

Table F. Populate Log Input Payload

Byte Offset	Length	Description
00h	10h	**Log Identifier:** UUID representing the log to populate.

Evaluation Copy