

OPEN

Compute Project

Hardware Error Reporting Standardization

Rev 0.4

Author(s): Rama Bhimanadhuni (MSFT), Zhengyu Yang (FB), Anil Agrawal(FB), Andrew Walton (Google) , Mark Fletcher (HPE), Yogesh Varma (Intel)...

1. License **(OCP CLA Option)**

Contributions to this Specification are made under the terms and conditions set forth in Open Compute Project Contribution License Agreement (“OCP CLA”) (“Contribution License”) by:

[Contributor Name(s) or Company name(s)]

Usage of this Specification is governed by the terms and conditions set forth in **[select one: Open Compute Project Hardware License – Permissive (“OCPHL Permissive”), Open Compute Project Hardware License – Copyleft (“OCPHL Reciprocal”)] (“Specification License”).**

Note: The following clarifications, which distinguish technology licensed in the Contribution License and/or Specification License from those technologies merely referenced (but not licensed), were accepted by the Incubation Committee of the OCP:

[insert “None” or a description of the applicable clarifications].

NOTWITHSTANDING THE FOREGOING LICENSES, THIS SPECIFICATION IS PROVIDED BY OCP "AS IS" AND OCP EXPRESSLY DISCLAIMS ANY WARRANTIES (EXPRESS, IMPLIED, OR OTHERWISE), INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR A PARTICULAR PURPOSE, OR TITLE, RELATED TO THE SPECIFICATION. NOTICE IS HEREBY GIVEN, THAT OTHER RIGHTS NOT GRANTED AS SET FORTH ABOVE, INCLUDING WITHOUT LIMITATION, RIGHTS OF THIRD PARTIES WHO DID NOT EXECUTE THE ABOVE LICENSES, MAY BE IMPLICATED BY THE IMPLEMENTATION OF OR COMPLIANCE WITH THIS SPECIFICATION. OCP IS NOT RESPONSIBLE FOR IDENTIFYING RIGHTS FOR WHICH A LICENSE MAY BE REQUIRED IN ORDER TO IMPLEMENT THIS SPECIFICATION. THE ENTIRE RISK AS TO IMPLEMENTING OR OTHERWISE USING THE SPECIFICATION IS ASSUMED BY YOU. IN NO EVENT WILL OCP BE LIABLE TO YOU FOR ANY MONETARY DAMAGES WITH RESPECT TO ANY CLAIMS RELATED TO, OR ARISING OUT OF YOUR USE OF THIS SPECIFICATION, INCLUDING BUT NOT LIMITED TO ANY LIABILITY FOR LOST PROFITS OR ANY CONSEQUENTIAL, INCIDENTAL, INDIRECT, SPECIAL OR PUNITIVE DAMAGES OF ANY CHARACTER FROM ANY CAUSES OF ACTION OF ANY KIND WITH RESPECT TO THIS SPECIFICATION, WHETHER BASED ON BREACH OF CONTRACT, TORT (INCLUDING NEGLIGENCE), OR OTHERWISE, AND EVEN IF OCP HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1. License **(OWF option)**

Contributions to this Specification are made under the terms and conditions set forth in Open Web Foundation Contributor License Agreement (“OWF CLA 1.0”) (“Contribution License”) by:

[Contributor Name(s) or Company name(s)]

Usage of this Specification is governed by the terms and conditions set forth in the Open Web Foundation Final Specification Agreement (“OWFa 1.0”).

Note: The following clarifications, which distinguish technology licensed in the Contribution License and/or Specification License from those technologies merely referenced (but not licensed), were accepted by the Incubation Committee of the OCP:

[insert “None” or a description of the applicable clarifications].

NOTWITHSTANDING THE FOREGOING LICENSES, THIS SPECIFICATION IS PROVIDED BY OCP "AS IS" AND OCP EXPRESSLY DISCLAIMS ANY WARRANTIES (EXPRESS, IMPLIED, OR OTHERWISE), INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR A PARTICULAR PURPOSE, OR TITLE, RELATED TO THE SPECIFICATION. NOTICE IS HEREBY GIVEN, THAT OTHER RIGHTS NOT GRANTED AS SET FORTH ABOVE, INCLUDING WITHOUT LIMITATION, RIGHTS OF THIRD PARTIES WHO DID NOT EXECUTE THE ABOVE LICENSES, MAY BE IMPLICATED BY THE IMPLEMENTATION OF OR COMPLIANCE WITH THIS SPECIFICATION. OCP IS NOT RESPONSIBLE FOR IDENTIFYING RIGHTS FOR WHICH A LICENSE MAY BE REQUIRED IN ORDER TO IMPLEMENT THIS SPECIFICATION. THE ENTIRE RISK AS TO IMPLEMENTING OR OTHERWISE USING THE SPECIFICATION IS ASSUMED BY YOU. IN NO EVENT WILL OCP BE LIABLE TO YOU FOR ANY MONETARY DAMAGES WITH RESPECT TO ANY CLAIMS RELATED TO, OR ARISING OUT OF YOUR USE OF THIS SPECIFICATION, INCLUDING BUT NOT LIMITED TO ANY LIABILITY FOR LOST PROFITS OR ANY CONSEQUENTIAL, INCIDENTAL, INDIRECT, SPECIAL OR PUNITIVE DAMAGES OF ANY CHARACTER FROM ANY CAUSES OF ACTION OF ANY KIND WITH RESPECT TO THIS SPECIFICATION, WHETHER BASED ON BREACH OF CONTRACT, TORT (INCLUDING NEGLIGENCE), OR OTHERWISE, AND EVEN IF OCP HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

2. Scope & Overview

Scope:

This document provides the Hardware Error reporting guidelines to be used for reporting the hardware errors handled by the firmware components like UEFI Firmware or BMC. It is recommended to design a consistent hardware error reporting methodology that can be adopted by OCP community.

Overview:

Hardware Error reporting formats currently used on the servers deployed in large clusters varies across various CSPs does not have use standardized error record formats. The error record format need to satisfy goals of identifying the FRU which caused the error, and provide extensive information about the error which can help identifying the first level cause of the error. In some cases, full error root cause may require hardware and software state.

Current error reporting formats is primarily driven by IPMI based SEL error reporting or UEFI spec based Common platform Error Record (CPER) error records used to describe the Errors as per ACPI based APEI tables (HEST, BERT etc). Currently firmware implementation uses IPMI based SEL error logs, ACPI based APEI tables or some custom implementations.

Even though CPER format are standardized in UEFI specification, firmware implementations do not use the CPER to report all errors across different implementations, and CPER records are not consumed widely in for lack of interface to read them out-of-band.

OxMs, CSPs use custom SEL log implementations to report different hardware errors as all hardware errors are not covered by the IPMI specification, and also SEL record is not sufficient provide all the details about the Hardware Error. This creates challenges in decoding these formats and requires integrating different vendor tools. These vendors tools integrating into Data center infrastructure provide security concerns. At the cloud scale this custom error log formats leads to different problems listed below

Cloud scale Diagnostics Issues	Impact
Lack of Clear failure signatures for Hardware errors.	Replacing multiple components leads to Increase in costs.
Lack of telemetry data required to root cause new Failures.	Contribute to long lead times for identifying mitigations and Solutions.

Lack of Standards-based Error log solutions.	<p>Custom error log formats -> Integrating Multiple Vendor Customs tools, which may not be cloud adoption ready.</p> <p>Leads to difficulty in Cloud based ML analytics for error pattern.</p>
--	---

3. Hardware Error Reporting requirements

Hardware errors can be reported by different agents like UEFI Firmware, BMC and OS based on the Error handling Implementation and type of hardware errors. Having the same error formats across different error reporting agents will be useful.

Currently Hardware error reporting uses two existing Standards based on errors to be reported as described below.

First one, IPMI specification-based System event logs (SEL) to report hardware errors. But IPMI based SEL have these limitations.

- SEL error records provide primitive data for reporting Hardware Errors.
- IPMI spec does not cover all types of Hardware errors e.g. lack of support for new technology based errors - CPU Interconnect errors (CXL, UPI, GDMI)
-
- Also, this IPMI specification is not maintained by the industry anymore, specification is becoming absolute.

In place of IPMI specification, DMTF organization has developed a secure and scalable interface specification called Redfish for the modern datacenter environments. This specification does not directly define hardware Error formats but provides an infrastructure to define the error records. There is need to define Hardware error standardization.

Second Industry standard is UEFI specification based Common platform Error Record (CPER) for Hardware error reporting. Currently UEFI firmware implementations use these CPER records for sharing the hardware error information to the OS. And OS also uses this CPER records to report Hardware errors handled by the operating system.

Hardware Error records access through BMC for Out of band purpose is very critical for fault diagnostics in both non bare metal and bare metal use cases. This is one of the widely used mechanism currently to harvest the error data reliably on scale of systems. Also, BMC based RAS error handling use cases is increasing which requires hardware error record format standardization and APIS to read these records.

4. Hardware Error Report Standardization Solutions

As IPMI specification-based System event logs (SEL) to report hardware errors have many limitations, having new standardized approach for Hardware Error records access through BMC with rich data about the hardware error is critical.

To have the Hardware Error standardization requires standardization in

1. Error Record format
2. Error Record Storage and
3. APIs to read the Error Records (In-band and Out-of-Band).

Error Report format standardization

Few solution options for Error Record format standardization

- Redfish Define Schema for Hardware Errors, this adds lot of complexity on the UEFI Firmware based SMI handlers, increases the latencies (List other reasons)
- CPER based error Records - As CPER based Hardware errors are already standardized as part of the UEFI specification adopting this solution for out-of-band method also gets a wider adoption.

CPER based error record is a recommend option for describing the error for all error reporting agents like UEFI firmware (BIOS), BMC and OS.

Error record Storage and APIs

Below table lists the different options used to describe the error, where to store the error and APIs used to read the errors.

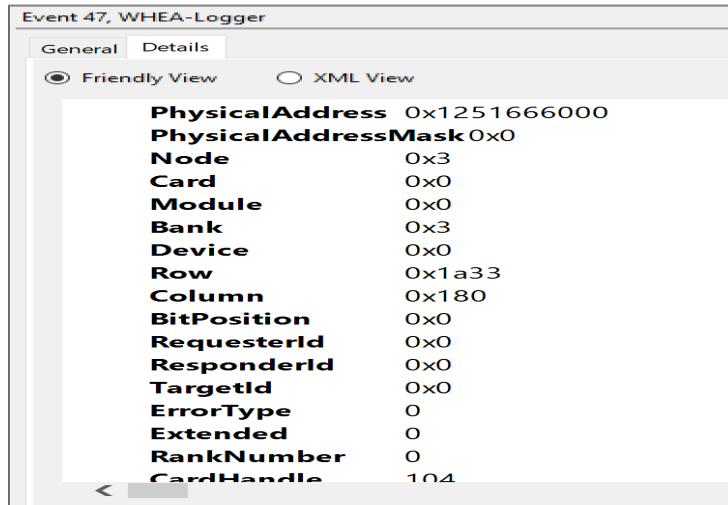
Error record format	Storage	Write API	Read API	Comments
SEL	BMC - NVRAM	ADD SEL Entry IPMI Command	"Get SEL Entry" IPMI Command	legacy
CPER In-band	BIOS - Flash	UEFI Variables Service	UEFI Variables Service	Provides Secure based access
CPER out-of-band	Platform specific implementation -BIOS based flash or BMC based flash.	Based on platform specific storage solution and security considerations APIs can differ.	Based on platform specific storage solution and security considerations APIs can differ.	Standardizing requires working with other OCP initiatives.

5. Appendix

5.1 Problem with IPMI based SEL reporting example

Problem example with IPMI SEL based error reporting, we see SEL has very limited data about the memory error, where as CPER based WHEA record provides the detailed description about the error.

```
sel.log - Notepad
File Edit Format View Help
ipmiutil ver 2.92
isel: version 2.92
-- BMC version 2.14, IPMI version 2.0
SEL Ver e3 Support 0f, Size = 227 records (Used=227, Free=0)
RecId Date/Time SEV Src Evt_Type Sens# Evt_detail - Trig [Evt_data]
0033 02/15/20 15:05:36 MAJ EFI Memory #87 Uncorrectable ECC, DIMM[11] 6f [a1 ff 0b]
0034 02/15/20 15:05:38 CRT SMI Processor #f3 Config error 6f [a5 92 11]
0035 02/15/20 15:05:38 TRF C 1/20 1/20 6f [01 ff ff]
```



5.2 Redfish compatibility for BMC Based error Handling.

Redfish provides the framework to define to send the binary blob of error data. Currently in DMTF redfish, we do not see **the standardization on error format, Error format schema is not defined yet**. It is up to the users to define the error format, it can lead to OXMS to define different formats. There is a need to standardize and define the redfish **Define Schema for all Hardware Error Types based on CPER**

Redfish log entry

https://redfish.dmtf.org/redfish/schema_index

https://redfish.dmtf.org/schemas/v1/LogEntry.v1_7_0.json

<https://redfish.dmtf.org/redfish/mockups/v1/953>

[redfish](#) » [v1](#) » [Systems](#) » [1](#) » [LogServices](#) » [SEL](#) » [Entries](#)

Example implementation for Memory Error in redfish Schema - with all the details (Initial version, Need to be revised)

```
redfish » v1 » Systems » 1 » LogServices » CPER » Entries » 2

"@odata.type": view details "#CPER - TBD",
"Id": view details "1",
"Name": view details "CPER Entry 1",
"EntryType": view details "CPER",
```

```

"Severity":view details "Critical",
"Created":view details "2012-10-27 T01:45:00Z",
"EntryCode":view details "TBD",
"Error Type":view details "Memory",
"Message":view details "Memory Uncorrectable Error",
"CPU": "0x1",
"Memory Controller": "0x1",
"Bank": "0x1",
"Rank": "0x0",
"Device": "0x1",
"Row": "0x1a33",
"Column": "0x180",
"Bit Position": "0x0",
"Physical address": "0x1251666000",
"Physical DIMM": "DIMM 7",

"MessageId":view details "0x592E28",
"Links":view details {
"OriginOfCondition":view details {
"@odata.id":view details "/redfish/v1/Chassis/1/CPER"go to this resource
} ,
"Oem":view details { }
} ,
"Oem":view details { } ,
"@odata.id":view
details "/redfish/v1/Systems/1/LogServices/CPER/Entries/2"

```

Dependency and Questions

1. If we define the error format, how do we influence the Industry to adopt.

- a. Once we define the format, should we work with DMTF Redfish forum to publish on the redfish as a reference.
- 2. In-band Redfish host Interface availability

5.3 Scope of Hardware Errors

These requirements cover for Hardware error types related to CPU, Memory, PCIe and Interconnects – CXL, Silicon Specific (CPU Internal Interconnects (UPI, xGMI etc).

- a. **Note.** Need to list any other errors we need to consider.