

GPU Operational Events (Preview)

Release 620

NVIDIA Corporation

Nov 11, 2026

Contents

1 Analyzing GPU Operational Events **3**
- 1.1 Overview . . . 3
- 1.2 Availability . . . 3
- 1.3 Event Identity . . . 4
- 1.4 Monitoring GPU Operational Events . . . 5
  - 1.4.1 Command Line . . . 5
  - 1.4.2 NVML . . . 5
  - 1.4.3 Kernel Log . . . 6
  - 1.4.4 Linux Tracepoints . . . 6
  - 1.4.5 Out-of-Band . . . 6
- 1.5 Interpreting an Event . . . 6
  - 1.5.1 Determine the Reporting Module and Event Code . . . 6
  - 1.5.2 Determine Impact from the Category and Severity . . . 6
  - 1.5.3 Read the Context Payloads . . . 7
  - 1.5.4 Determine the Actions to Take . . . 7

2 GPU Operational Event Formats **9**
- 2.1 Structured Event Records . . . 9
- 2.2 Linux Tracepoints . . . 12
- 2.3 UEFI CPER . . . 12
- 2.4 Legacy Xid Kernel Log Lines . . . 13

3 GPU Operational Event Catalog **15**
- 3.1 Event Categories . . . 15
- 3.2 Event Log Levels . . . 17
- 3.3 Event Severities . . . 17
- 3.4 Event Scopes . . . 17
- 3.5 Published Events . . . 18
  - 3.5.1 Event Lookup . . . 18
  - 3.5.2 Event Details . . . 24
    - 3.5.2.1 GPU . . . 24
    - 3.5.2.2 GPU-BUS . . . 25
    - 3.5.2.3 GPU-FSP . . . 25
    - 3.5.2.4 GPU-GSP . . . 26
    - 3.5.2.5 GPU-MEMSYS . . . 28
    - 3.5.2.6 GPU-NVLINK . . . 28
    - 3.5.2.7 GPU-PCIE . . . 43
    - 3.5.2.8 GPU-PSHC . . . 44
    - 3.5.2.9 GPU-RAS . . . 45
    - 3.5.2.10 GPU-RC . . . 67

4 Release Notes **69**
- 4.1 New Event Coverage . . . 69

4.2 New Reporting Capabilities 69

5 Notices **71**
5.1 Trademarks 72

Warning

GPU Operational Events are a **preview feature** and remain under active development. Category values, module signatures, module event codes, severities, context payloads, and the set of published events may change before the interface is declared stable. Treat the identifiers in this document as valid for the driver release documented here, and re-validate consumers against each driver release. Legacy Xid reporting remains supported for production monitoring during the preview period. See *Release Notes* for compatibility changes.

This document explains what GPU Operational Events are, how to monitor them, the formats they are published in, and the published event categories and codes. It is intended to assist system administrators, developers, and FAEs who consume structured GPU health and error data from the NVIDIA driver.

This edition documents **R620**. See *Release Notes* for changes from R615.

Chapter 1. Analyzing GPU Operational Events

Warning

GPU Operational Events are a **preview feature** and remain under active development. Category values, module signatures, module event codes, severities, context payloads, and the set of published events may change before the interface is declared stable. Treat the identifiers in this document as valid for the driver release documented here, and re-validate consumers against each driver release. Legacy Xid reporting remains supported for production monitoring during the preview period. See *Release Notes* for compatibility changes.

1.1. Overview

A **GPU Operational Event (GOE)** is a structured, machine-readable record that the NVIDIA GPU driver publishes whenever it detects a condition worth reporting to a system operator: an error, a recovery action, a resource retirement, or a notable state change. Each event carries an identity, a severity, an impact scope, and a list of typed context payloads that describe the specific occurrence.

GPU Operational Events give system software a programmatic interface for the same conditions that the driver reports as Xid messages in the kernel log. Each event carries its identity in discrete numeric fields.

Events that have a legacy Xid equivalent continue to produce `NVRM: Xid (PCI:...)` kernel log lines. The structured event and the legacy Xid line describe the same occurrence. See *Release Notes* for changes to individual Xid messages.

For the actions to take in response to a condition, see the Xid Errors documentation. Each event in *Published Events* lists its legacy Xid, which selects the corresponding entry in the Xid catalog and its recommended immediate and investigatory actions.

1.2. Availability

This document describes GPU Operational Events in R620. NVML support is available on Linux, on Turing and later GPUs. Structured subscriptions, retained CPER queries, and the `nvidia-smi` commands below require administrative privileges (`CAP_SYS_ADMIN` on Linux).

1.3. Event Identity

Event identity consists of a module signature, category, and module event code.

Field	Type	Description
Module Signature	Character array	ASCII string naming the source module, for example GPU-GSP. Namespaces the event code.
Category	16-bit integer	Class of condition, from the category registry in *Event Categories*. Shared across all modules.
Module Event Code	16-bit integer	Specific event within the reporting module. Together with the module signature and category, locates matching entries in *Published Events*.

Each event additionally carries the fields below.

Field	Description
Severity	Impact relative to the event scope. See *Event Severities*.
Log Level	Urgency of the report, orthogonal to severity. See *Event Log Levels*.
Scope	What the event affects. See *Event Scopes*.
Originator	Which driver or firmware component originated the event.
Device UUID	UUID of the GPU that reported the event.
Instance ID	Originator-scoped sequence number.
Timestamp	Microsecond wall clock.
Trace ID	Correlation identifier shared by related events that describe one incident.
Group Cursor	64-bit value that identifies the event group. Also maps to the CPER Error Record Header `recordId` for the retained CPER record.
Group Position	Zero-based index within the group visible to the subscriber.
Group Size	Number of events in the group visible to the subscriber.
Attributes	Per-event flags that describe properties of the event.
Group Attributes	Group-level flags that describe properties of the event group.
Attribution	Fields for locating the event within the device. R620 does not report MIG instance IDs for GPU Operational Events.

The trace ID correlates related events in a causal chain; the group cursor identifies one exported group. NVML delivers one event per successful wait. Use `groupCursor` to associate those records, `groupIndex` to order them, and `groupSize` to check completeness. See *GPU Operational Event Formats* for how these fields appear in each published format.

1.4. Monitoring GPU Operational Events

The driver retains recent event groups in an in-memory Operational Event log. The interfaces below expose event records for monitoring and diagnosis.

1.4.1. Command Line

`nvidia-smi event-log` prints retained events since driver load in a human-readable form. It reports events for every GPU visible to NVML, skipping inaccessible GPUs during enumeration. `event-log` accepts `-h` / `--help`. Resolve the numeric values it prints through *Published Events*. The decoder has the following limitations in this release:

- Module event codes print as hexadecimal values without names.
- Categories `0x0008` and `0x0009` still print as `Engine Software Error` and `Engine Hardware Error`. In R620 these mean **Firmware Fault** and **Engine Error**, respectively. Use the numeric category and this release's catalog.
- Contexts other than Legacy Xid print their numeric type and no payload. Legacy Xid contexts print the Xid code.
- `Severity` reports the CPER severity of the record rather than the event's own severity, so a GPU-fatal event appears here as `Recoverable`. See *UEFI CPER* for the mapping, and read the severity in a structured record for GPU-relative impact.

`nvidia-smi cper` prints retained history as Base64-encoded raw CPER records, one record per Base64 block, for forwarding to an external collector. `cper` accepts `-h` / `--help` and `--no-wrap`. Output wraps at 76 columns by default; `--no-wrap` emits each CPER record on a single line.

1.4.2. NVML

NVML exposes GPU Operational Events through two interfaces. See the NVML API Reference Guide for full function prototypes.

Structured event subscription. Register a GPU with `nvmlEventSetRegisterGpuOperationalEvents_v1`, then retrieve events with `nvmlEventSetWait_v3`. Records with `dataType` set to `NVML_EVENT_DATA_TYPE_GPU_OPERATIONAL_EVENT` carry the structured event. Read attached context payloads with `nvmlEventSetGetContextCount_v1`, `nvmlEventSetGetContextInfo_v1`, and `nvmlEventSetGetContextData_v1`. Decode a legacy Xid context with `nvmlEventSetGetGpuOperationalEventContextLegacyXid_v1`.

Zero-initialize `nvmlGpuOperationalEventConfig_v1_t` and set its `uuid` to the target `GPU-...` UUID. This subscribes to new device-wide events after registration. Optional `minLogLevel` and `minSeverity` thresholds filter whole groups; zero disables each filter. These thresholds do not remove lower-level companion events from a matching group.

Read or copy contexts before the next successful `nvmlEventSetWait_v3` on that event set: the context accessors refer to the last returned event, not a caller-owned event copy. The typed Legacy Xid accessor returns the Xid code; use the raw context for its message.

The same event set can subscribe to legacy Xid events and GPU Operational Events. When a GPU Operational Event carries a Legacy Xid context, use that Xid value as a correlation hint to the matching Xid event, kernel log line, and catalog entry.

Retained CPER history. `nvmlSystemGetCPER_v1` returns retained CPER records as raw bytes. This is the interface that both `nvidia-smi` commands above use. Initialize the cursor handle to

NVML_CPER_CURSOR_HANDLE_INIT and preserve it between calls. A size query uses a null buffer and zero buffer size; NVML_ERROR_INSUFFICIENT_SIZE returns the required size. A successful call with zero buffer size means no more records.

1.4.3. Kernel Log

Events that carry a legacy Xid context emit their traditional kernel log line:

```
NVRM: Xid (PCI:0000:41:00): 63, Row Remapper: New row (0x00000000fbc12000)
↪marked for remapping, reset gpu to activate.
```

The Xid prefix and numeric code remain available to existing log parsers. Simulated events add [SIMULATED] to the message. Events without a legacy Xid equivalent are visible through the structured, CPER, and out-of-band interfaces.

1.4.4. Linux Tracepoints

The open kernel module exposes five tracepoints in the nvidia tracing subsystem: nvidia_gpu_error, nvidia_gpu_warning, nvidia_gpu_notice, nvidia_gpu_diagnostic, and nvidia_gpu_telemetry. Each event goes to the tracepoint for its log level. They carry event identity, severity, scope, attribution, and correlation fields; context payloads remain available through NVML and CPER.

With tracefs mounted at /sys/kernel/tracing, run as root to capture error-level events:

```
echo 1 > /sys/kernel/tracing/events/nvidia/nvidia_gpu_error/enable
cat /sys/kernel/tracing/trace_pipe
```

After stopping the capture, write 0 to the same enable file. These tracepoints are unavailable with the proprietary kernel module.

1.4.5. Out-of-Band

On platforms that support out-of-band reporting, the driver forwards rendered CPER records to the BMC. Match the CPER record ID to the structured event's group cursor to correlate reports.

1.5. Interpreting an Event

1.5.1. Determine the Reporting Module and Event Code

Read the module signature and module event code from the structured record, or Source Module and Module Event Code from nvidia-smi event-log. Use the category and context type to find the matching entry in *Published Events*.

1.5.2. Determine Impact from the Category and Severity

The category identifies the class of condition and the severity identifies impact relative to the event's scope. Use scope to establish what the severity applies to: a fatal event at individual GPU engine scope reports that one engine is down, while a fatal event at device scope reports that the GPU is down.

1.5.3. Read the Context Payloads

Contexts carry occurrence-specific data. In this release, NVML and `nvidia-smi` decode the Legacy Xid code. NVML also exposes raw payloads, including the Legacy Xid message and other typed contexts. Check the source context type, format version, and size before decoding; skip unrecognized payloads using their size. See *GPU Operational Event Formats* for the context layouts.

1.5.4. Determine the Actions to Take

For an event that lists a legacy Xid in *Published Events*, look the Xid up in the Xid Errors catalog, which supplies the immediate and investigatory actions for that condition. For `LTS_REPAIR_PENDING` and `MEMORY_CHANNEL_REPAIR_PENDING`, the Xid 160 message states the action needed to activate the repair.

Chapter 2. GPU Operational Event Formats

Warning

GPU Operational Events are a **preview feature** and remain under active development. Category values, module signatures, module event codes, severities, context payloads, and the set of published events may change before the interface is declared stable. Treat the identifiers in this document as valid for the driver release documented here, and re-validate consumers against each driver release. Legacy Xid reporting remains supported for production monitoring during the preview period. See *Release Notes* for compatibility changes.

GPU Operational Events are published in several formats. These describe the same underlying events, but the context data available in each format can differ.

Format	Typical Consumers
Structured Event Records	NVML subscribers
UEFI CPER	BMC collection, `nvidia-smi event-log`, `nvidia-smi cper`, and `nvmlSystemGetCPER_v1`
Linux Tracepoints	Kernel tracing with the open GPU kernel modules
Legacy Xid Kernel Log Lines	Existing `dmesg` and syslog parsers

See *Analyzing GPU Operational Events* for how to monitor events and how to interpret categories and published codes.

2.1. Structured Event Records

Structured records expose the event identity and attribution as discrete fields. Through NVML, a successful `nvmlEventSetWait_v3` returns records whose `dataType` is `NVML_EVENT_DATA_TYPE_GPU_OPERATIONAL_EVENT`. Those records carry:

- Module signature, category, and module event code
- Severity, log level, scope, and originator

- Device UUID and attribution
- Trace ID, group cursor, group position, and instance ID
- Timestamp and per-event / group attributes

`groupSize` counts the events visible to the subscriber's scope, and `groupIndex` is their zero-based position within that group.

Context payloads follow the event. Each context has a type, a format version, and a payload size.

Domain-independent types carry generically decodable data:

Type	Contents
`0x0000`	Opaque bytes
`0x0001`	Key/value pairs, 64-bit values
`0x0002`	Key/value pairs, 32-bit values
`0x0003`	Value array, 64-bit values
`0x0004`	Value array, 32-bit values

GPU domain types begin at `0x8001`; `0x8000` is reserved. The types defined in this release are:

Type	Name	Contents
`0x8001`	Legacy Xid	Legacy Xid code and formatted message
`0x9001`	GPU Timeout Data	Timeout-specific fields such as waited duration
`0xA001`	GPU Initialization Metadata	Driver and device metadata reported at initialization
`0xA101`	GSP RPC Timeout Data	RPC function, name, sequence number, and wait duration
`0xA201`	FSP Boot Timeout Data	FSP boot polling state
`0xA202`	FSP Fuse Error Data	FSP fuse status
`0xA301`	NVLink ALI Training Failure Data	Affected link IDs and debug registers
`0xA302`	NVLink MSE Error Data	MSE error status and debug registers
`0xA303`	NVLink Software Link-Down Data	Link ID and eight unused debug words
`0xA304`	NVLink NVL3 Fatal Error Data	Link ID and seven debug words
`0xA305`	NVLink Privilege Error Data	Link ID
`0xA306`	NVLink Blackwell Error Data	Link ID (-1 if not link-specific), error status, interrupt information, and eight debug words
`0xA307`	NVLink SAW Error Data	64 reserved bytes
`0xA308`	NVLink RLW Error Data	64 reserved bytes
`0xA309`	NVLink TLW Error Data	64 reserved bytes

continues on next page

Table 1 – continued from previous page

Type	Name	Contents
0xA30A	NVLink TREX Error Data	64 reserved bytes
0xA30B	NVLink NVLPW Control Error Data	64 reserved bytes
0xA30C	NVLink NETIR Error Data	64 reserved bytes
0xA501	Secure NVLink IR2SAW Interrupt Data	Secure NVLink error fields
0xA502	Secure NVLink Cross-Containment Data	Secure NVLink error fields
0xA503	Secure NVLink Encryption Error Data	Secure NVLink error fields
0xA504	Secure NVLink Remote Hardware Error Data	Secure NVLink error fields
0xA505	Secure NVLink Local STO Error Data	Secure NVLink error fields
0xA506	Secure NVLink Local Privilege Error Data	Secure NVLink error fields
0xA507	Secure NVLink Remote Privilege Error Data	Secure NVLink error fields
0xA508	Secure NVLink DMEM Parity Error Data	Secure NVLink error fields
0xA509	Secure NVLink IMEM Parity Error Data	Secure NVLink error fields
0xA50A	Secure NVLink MPU Parity Error Data	Secure NVLink error fields
0xA50B	Secure NVLink Register Parity Error Data	Secure NVLink error fields
0xA50C	Secure NVLink Traffic Error Data	Secure NVLink error fields
0xA50D	Secure NVLink Instruction Cache Parity Error Data	Secure NVLink error fields
0xA50E	Secure NVLink Data Cache Parity Error Data	Secure NVLink error fields
0xA601	PCIe Fatal Timeout Data	Timeout source and register value
0xA602	PCIe Correctable Error Fault Data	Correctable-error rate tuple number, NERD number, and tuple version
0xA603	PCIe Reset Failure Data	Reset status and recommended recovery action

Private contexts use opaque type `0x0000` and do not have a published field layout.

`nvidia-smi event-log` prints structured events in human-readable form.

2.2. Linux Tracepoints

R620 adds five tracepoints in the `nvidia` trace system: `nvidia_gpu_error`, `nvidia_gpu_warning`, `nvidia_gpu_notice`, `nvidia_gpu_diagnostic`, and `nvidia_gpu_telemetry`. The event's log level selects the tracepoint. These are available with the open GPU kernel modules.

The raw trace record carries the PCI device, UUID, module signature and event code, category, severity, scope, originator, event and group attributes, log level, trace ID, instance ID, timestamp, and module, chiplet, and MIG attribution. It does not carry contexts or group cursor, size, or index. The formatted trace line shows a subset of those fields.

2.3. UEFI CPER

GPU Operational Events are encoded as UEFI Common Platform Error Records. See Appendix N of the UEFI Specification for the CPER format. CPER is the format used for out-of-band delivery to a BMC and for retained history access through `nvmlSystemGetCPER_v1` and `nvidia-smi cper`.

One CPER record represents one event group. Each event in the group becomes one section of type **NVIDIA Event Section**, GUID `9068e568-6ca0-11f0-aeaf-159343591eac`. The Creator ID and Notification Type GUIDs listed below are version-5 GUIDs derived from the NVIDIA namespace `ea219640-18b9-4c91-8e5c-58eeacd9b33f`. Match these GUIDs by value.

The NVIDIA Event Section begins with a 32-byte common header, followed by a device-type-specific event info structure, followed by the context list. The event identity maps as follows:

CPER Field	Source	Notes
Source Device Type	GPU (1)	Selects the event info layout.
Event Type (u16)	Category	Value from the category registry.
Event Sub-type (u16)	Module Event Code	Specific event within the reporting module.
Source Module Signature (`char[16]`)	Module Signature	NUL-terminated ASCII string naming the source module.
Event Trace ID (u64)	Trace ID	Full 64-bit value including the originator prefix.
Error Record Header `recordId`	Group Cursor	The driver writes the group cursor into `recordId`, so the same value identifies the event group in both the structured record and the CPER record.

Event info for the GPU device type is 16 bytes, version 1.0 (`0x0100`). It adds the event originator, the device PDI, and source partition and subpartition fields reserved for future use. Each context has a 16-byte header, version 0.0, and is padded to 16-byte alignment. `contextSize` includes the header and padding; `dataSize` counts only the payload bytes.

CPER uses the same context type registry, but context encoding depends on the reporting path. Native event serializers encode most module-specific contexts as opaque type `0x0000`, version 0. Legacy Xid and initialization metadata retain their typed encodings (`0x8001` and `0xA001`, version 1.0). Native PCIe correctable-error serialization emits the event section but omits its correctable-error and Legacy Xid contexts.

When the host imports a firmware event, CPER preserves the context types and versions from the imported structured record. A module context can therefore be typed on this path and opaque on another. Decode each context using its own header; use structured records for the published module-specific layouts when CPER supplies opaque data.

The Error Record Header identifies the producer through a **Creator ID** and describes how the condition was discovered through a **Notification Type**:

Role	GUID	Meaning
Creator ID	`5a572ae4-9a04-5e55-a86f-db`	Record created by the host driver on the physical function.
Notification Type	`0eeaf365-3740-5271-ae58-10`	Reported through a GPU hardware interrupt.
Notification Type	`e0fd0b2d-ba62-59c1-9744-9d`	Detected by on-device firmware fault handling.
Notification Type	`3a90eac6-481d-5cd7-b57d-0a`	Inferred from a timeout waiting on GPU hardware or firmware.
Notification Type	`cb2668a9-0508-5e36-bbc9-4d`	Detected by a software validation or invariant check.

GOE Severity to CPER Severity Mapping

GOE severity is evaluated at the event's explicit scope, such as a device or engine. CPER severity uses the UEFI encoding — recoverable (0), fatal (1), corrected (2), informational (3) — and is always evaluated at the host scope. Encoding a GOE as CPER therefore re-scopes the impact to the host, so the CPER severity can differ from the GOE severity.

The driver maps GOE fatal and GOE recoverable to CPER recoverable, GOE corrected to CPER corrected, and GOE informational to CPER informational. GPU-fatal conditions leave the host recoverable, so the driver emits CPER recoverable for them and does not emit CPER fatal. The NVIDIA Event Section carries identity and attribution fields, while GOE severity and scope remain on the structured event record. For GPU-relative impact, read those fields from the structured record; they are not recoverable from the CPER severity alone.

Group attributes map to Error Record Header flags: recovered, previous error, and simulated.

2.4. Legacy Xid Kernel Log Lines

Contexts of type `0x8001` carry a legacy Xid code and its formatted message. These produce the traditional kernel log line:

```
NVRM: Xid (PCI:0000:41:00): 63, Row Remapper: New row (0x00000000fbc12000)
↪marked for remapping, reset gpu to activate.
```

Most events that report an Xid also deliver that code and message as a `0x8001` context inside structured records and CPER, so a programmatic consumer may read them without parsing `dmesg`.

Chapter 3. GPU Operational Event Catalog

Warning

GPU Operational Events are a **preview feature** and remain under active development. Category values, module signatures, module event codes, severities, context payloads, and the set of published events may change before the interface is declared stable. Treat the identifiers in this document as valid for the driver release documented here, and re-validate consumers against each driver release. Legacy Xid reporting remains supported for production monitoring during the preview period. See *Release Notes* for compatibility changes.

This page lists the shared event category registry and the GPU Operational Events published in R620. See *Analyzing GPU Operational Events* for monitoring interfaces and how to interpret an event, and *GPU Operational Event Formats* for the formats events are published in.

3.1. Event Categories

The category registry is shared by all reporting modules. A consumer can act on the category alone when it does not recognize a specific module event code.

Category	Value	Indicates
TIMEOUT	0x0001	An operation did not complete within its expected time budget.
MEMORY_INTEGRITY_ERROR	0x0002	A data integrity error in memory or cache, such as ECC, parity, or poison.
INTERCONNECT_ERROR	0x0003	An error on a link between the GPU and another device, such as NVLink, PCIe, or C2C.
CORRUPTION_DETECTED	0x0004	Corrupted driver or device state was detected.
SECURITY_ERROR	0x0005	A security policy violation or an authentication failure.
POWER_ERROR	0x0006	A power delivery or power management fault.
THERMAL_ERROR	0x0007	A thermal limit was reached or a thermal sensor failed.
FIRMWARE_FAULT	0x0008	A fault detected by firmware running on the device.
ENGINE_ERROR	0x0009	An error reported by a GPU engine.
RESOURCE_EXHAUSTED	0x000A	A repair, spare, or tracking resource has been used up.
TRANSLATION_FAULT	0x000B	An address translation or MMU fault.
UNCLASSIFIED_ERROR	0x7FFF	An error that has not yet been assigned a specific category.
INITIALIZATION	0x8000	Component initialization progress or completion.
SHUTDOWN	0x8001	Component shutdown or teardown.
STATE_CHANGED	0x8002	An operational state transition.
CONFIG_CHANGED	0x8003	A configuration change.
RESOURCE_RETIREMENT	0x8004	A resource has been marked for retirement or repair.
RESOURCE_THRESHOLD	0x8005	A resource usage threshold was crossed.
TELEMETRY	0xFFFE	Operational telemetry.
DEBUG	0xFFFF	Engineering and debug data.

This release publishes events in ten of these categories: TIMEOUT, MEMORY_INTEGRITY_ERROR, INTERCONNECT_ERROR, FIRMWARE_FAULT, RESOURCE_EXHAUSTED, UNCLASSIFIED_ERROR, INITIALIZATION, STATE_CHANGED, RESOURCE_RETIREMENT, and RESOURCE_THRESHOLD. The remaining values are reserved for other event categories.

Note

In this release, the `nvidia-smi event-log` decoder renders category `0x0008` as "Engine Software Error" and `0x0009` as "Engine Hardware Error". These values correspond to FIRMWARE_FAULT and ENGINE_ERROR, respectively. Decode categories from the numeric value.

3.2. Event Log Levels

The log level describes the urgency of the report independently from the event severity. Values increase with urgency, so a subscription filter can use "at least this urgent" semantics. A filter value of 0 matches every log level.

Log Level	Value	Indicates
`TELEMETRY`	`10`	Operational telemetry.
`DIAGNOSTIC`	`20`	Diagnostic information. NVML names this level `NVML_GPU_OPERATIONAL_EVENT_LOG_LEVEL_DIAG`.
`NOTICE`	`30`	A notable condition that does not require immediate action.
`WARNING`	`40`	A condition that may require operator attention.
`ERROR`	`50`	A condition that requires operator attention.

3.3. Event Severities

The severity identifies the impact of the event relative to its scope. Values increase with severity, so a subscription filter can use "at least this severe" semantics. A filter value of 0 matches every severity.

Severity	Value	Indicates
`INFORMATIONAL`	`10`	The event reports state, telemetry, or progress.
`CORRECTED`	`20`	The condition was corrected without recovery action by the operator.
`RECOVERABLE`	`30`	The condition affected operation, and recovery is possible.
`FATAL`	`40`	The condition caused loss of the scoped resource.

When a GPU Operational Event is encoded as CPER, the GOE severity is remapped to a host-scoped CPER severity. See *GOE Severity to CPER Severity Mapping*.

3.4. Event Scopes

The scope identifies the resource affected by the event.

Scope	Value	Indicates
SYSTEM	0	The whole system.
DEVICE	1	One GPU device.
GPU_INSTANCE	2	One MIG GPU instance.
COMPUTE_INSTANCE	3	One MIG compute instance.
ENGINE	4	One GPU engine.

3.5. Published Events

This release publishes the events listed below. To identify an event you have decoded, find its module signature and module event code in the lookup table, then follow the event name to its entry. Not all Xids emitted by the driver have been fully converted to GPU Operational Events yet; more support will be added in future releases. To determine the actions to take, look up the event's legacy Xid in the Xid Errors catalog.

Where severity is shown as FATAL or RECOVERABLE, the driver selects the severity at runtime according to the impact of the reported condition. Some events emit more than one legacy Xid; these are listed together. Event-specific context may be opaque, depending on the output format.

3.5.1. Event Lookup

Module	Code	Event	Legacy Xid	Category
GPU	0x0001	*DRIVER_INIT*	None	INITIAL-IZATION
GPU-BUS	0x0002	*C2C_CONTAINMENT*	179	INTERCON-NECT_ERROR
GPU-BUS	0x0003	*GPU_FALLEN_OFF_BUS*	79	INTERCON-NECT_ERROR
GPU-FSP	0x0001	*BOOT_TIMEOUT*	143	TIMEOUT
GPU-FSP	0x0002	*FUSE_ERROR*	143	FIRMWARE_FAULT
GPU-GSP	0x0001	*RPC_TIMEOUT*	119	TIMEOUT
GPU-GSP	0x0002	*LIBOS_HEARTBEAT_TIMEOUT*	120	TIMEOUT
GPU-GSP	0x0003	*GSP_RM_HEARTBEAT_TIMEOUT*	120	TIMEOUT
GPU-GSP	0x0004	*FIRMWARE_FAULT*	120	FIRMWARE_FAULT
GPU-GSP	0x0005	*POISON*	140	MEM-ORY_INTEGRITY_ERROR

continues on next page

Table 2 – continued from previous page

Module	Code	Event	Legacy Xid	Category
GPU-MEMSYS	0x0001	*MAINT_OP_ERROR*	175	MEMORY_INTEGRITY_ERROR
GPU-NVLINK	0x0001	*ALI_TRAINING_FAILURE*	136	INTERCONNECT_ERROR
GPU-NVLINK	0x0001	*IR2SAW_INTERRUPT*	170	INTERCONNECT_ERROR
GPU-NVLINK	0x0002	*MSE_DEGRADED*	150	FIRMWARE_FAULT
GPU-NVLINK	0x0002	*CROSS_CONTAIN*	170	INTERCONNECT_ERROR
GPU-NVLINK	0x0003	*MSE_WATCHDOG_TIMEOUT*	150	FIRMWARE_FAULT
GPU-NVLINK	0x0003	*ENCRYPT_ERROR*	170	INTERCONNECT_ERROR
GPU-NVLINK	0x0004	*SW_LINK_DOWN*	155	INTERCONNECT_ERROR
GPU-NVLINK	0x0004	*REMOTE_HW_ERR*	170	INTERCONNECT_ERROR
GPU-NVLINK	0x0005	*FATAL_ERROR_NVL3*	74	INTERCONNECT_ERROR
GPU-NVLINK	0x0005	*LOCAL_STO_ERR*	170	INTERCONNECT_ERROR
GPU-NVLINK	0x0006	*PRIV_ERROR*	137	INTERCONNECT_ERROR
GPU-NVLINK	0x0006	*LOCAL_PRIV_ERR*	170	INTERCONNECT_ERROR
GPU-NVLINK	0x0007	*SAW_ERROR_GPU_FATAL*	144	INTERCONNECT_ERROR
GPU-NVLINK	0x0007	*REMOTE_PRIV_ERR*	170	INTERCONNECT_ERROR
GPU-NVLINK	0x0008	*SAW_ERROR_NON_FATAL_WARNING*	144	INTERCONNECT_ERROR
GPU-NVLINK	0x0008	*DMEM_PARITY_ERR*	170	INTERCONNECT_ERROR
GPU-NVLINK	0x0009	*SAW_ERROR_NON_FATAL_CONTAINED*	144	INTERCONNECT_ERROR
GPU-NVLINK	0x0009	*IMEM_PARITY_ERR*	170	INTERCONNECT_ERROR
GPU-NVLINK	0x000A	*RLW_ERROR_GPU_FATAL*	145	INTERCONNECT_ERROR

continues on next page

Table 2 – continued from previous page

Module	Code	Event	Legacy Xid	Category
GPU-NVLINK	0x000A	*MPU_PARITY_ERR*	170	INTERCON-NECT_ERROR
GPU-NVLINK	0x000B	*RLW_ERROR_NON_FATAL_WARNING*	145	INTERCON-NECT_ERROR
GPU-NVLINK	0x000B	*REG_PARITY_ERR*	170	INTERCON-NECT_ERROR
GPU-NVLINK	0x000C	*RLW_ERROR_APPLICATION_FATAL*	145	INTERCON-NECT_ERROR
GPU-NVLINK	0x000C	*TRAFFIC_ERR*	170	INTERCON-NECT_ERROR
GPU-NVLINK	0x000D	*TLW_ERROR_GPU_FATAL*	146	INTERCON-NECT_ERROR
GPU-NVLINK	0x000D	*ICACHE_PARITY_ERR*	170	INTERCON-NECT_ERROR
GPU-NVLINK	0x000E	*TLW_ERROR_NON_FATAL_WARNING*	146	INTERCON-NECT_ERROR
GPU-NVLINK	0x000E	*DCACHE_PARITY_ERR*	170	INTERCON-NECT_ERROR
GPU-NVLINK	0x000F	*TLW_ERROR_NON_FATAL_CONTAINED*	146	INTERCON-NECT_ERROR
GPU-NVLINK	0x000F	*IR2SAW_RECOVERABLE*	170	INTERCON-NECT_ERROR
GPU-NVLINK	0x0010	*TREX_ERROR_NON_FATAL_WARNING*	147	INTERCON-NECT_ERROR
GPU-NVLINK	0x0010	*CROSS_CONTAIN_RECOVERABLE*	170	INTERCON-NECT_ERROR
GPU-NVLINK	0x0011	*NVLPW_ERROR_NON_FATAL_WARNING*	148	INTERCON-NECT_ERROR
GPU-NVLINK	0x0011	*LOCAL_STO_ERR_RECOVERABLE*	170	INTERCON-NECT_ERROR
GPU-NVLINK	0x0012	*NETIR_ERROR_NON_FATAL_WARNING*	149	INTERCON-NECT_ERROR
GPU-NVLINK	0x0013	*NETIR_ERROR_GPU_FATAL*	149	INTERCON-NECT_ERROR
GPU-NVLINK	0x0014	*NETIR_ERROR_LINK_FATAL*	149	INTERCON-NECT_ERROR
GPU-PCIE	0x0001	*FATAL_TIMEOUT*	167	INTERCON-NECT_ERROR

continues on next page

Table 2 – continued from previous page

Module	Code	Event	Legacy Xid	Category
GPU-PCIE	0x0002	*CORRECTABLE_ERROR_FAULT*	176	INTERCONNECT_ERROR
GPU-PCIE	0x0003	*RESET_FAILURE*	182	INTERCONNECT_ERROR
GPU-PSHC	0x0001	*PSHC_REENGAGED*	162	STATE_CHANGED
GPU-PSHC	0x0002	*PSHC_DISENGAGED*	163	STATE_CHANGED
GPU-PSHC	0x0003	*PSHC_LOW_LIFETIME*	164	RESOURCE_THRESHOLD
GPU-PSHC	0x0004	*PSHC_ZERO_LIFETIME*	165	RESOURCE_THRESHOLD
GPU-RAS	0x0001	*ROW_REMAPPING_PENDING*	63	RESOURCE_RETIREMENT
GPU-RAS	0x0002	*ROW_REMAPPING_FAILURE_TABLE_FULL*	64	RESOURCE_EXHAUSTED
GPU-RAS	0x0003	*ROW_REMAPPING_FAILURE_BANK_FULL*	64	RESOURCE_EXHAUSTED
GPU-RAS	0x0004	*ROW_REMAPPING_FAILURE_INTERNAL_ERROR*	64	UNCLASSIFIED_ERROR
GPU-RAS	0x0005	*ROW_REMAPPING_FAILURE_RESERVED_ROW*	64	RESOURCE_EXHAUSTED
GPU-RAS	0x0006	*ROW_REMAPPING_FAILURE_PAGE_OFFLINE_FA*	64	UNCLASSIFIED_ERROR
GPU-RAS	0x0007	*DRAM_RETIREMENT_FAILURE_INTERNAL_ERR(*	64	UNCLASSIFIED_ERROR
GPU-RAS	0x0008	*DRAM_RETIREMENT_FAILURE_INFOROM_FULL*	64	RESOURCE_EXHAUSTED
GPU-RAS	0x0009	*DRAM_RETIREMENT_FAILURE_HW_LIMIT*	64	RESOURCE_EXHAUSTED
GPU-RAS	0x000A	*DRAM_RETIREMENT_FAILURE_NO_SPARE*	64	RESOURCE_EXHAUSTED
GPU-RAS	0x000B	*LTS_REPAIR_PENDING*	160	RESOURCE_RETIREMENT
GPU-RAS	0x000C	*LTS_REPAIR_FAILURE*	161	RESOURCE_EXHAUSTED
GPU-RAS	0x000D	*MEMORY_CHANNEL_REPAIR_PENDING*	160	RESOURCE_RETIREMENT
GPU-RAS	0x000E	*MEMORY_CHANNEL_REPAIR_FAILURE*	161	RESOURCE_EXHAUSTED

continues on next page

Table 2 – continued from previous page

Module	Code	Event	Legacy Xid	Category
GPU-RAS	0x000F	*SPARING_REPAIR_PENDING*	156	RE-SOURCE_RETIREMENT
GPU-RAS	0x0011	*SPARING_REPAIR_NOT_POSSIBLE*	157	RE-SOURCE_EXHAUSTED
GPU-RAS	0x0013	*ECC_RESIDUAL_UNCORRECTABLE_ERROR*	140	MEM-ORY_INTEGRITY_ERROR
GPU-RAS	0x0014	*DRAM_ECC_INTR_STORM*	92	MEM-ORY_INTEGRITY_ERROR
GPU-RAS	0x0015	*SM_ECC_INTR_STORM*	92	MEM-ORY_INTEGRITY_ERROR
GPU-RAS	0x0016	*BANK_REMAPPING_PENDING*	177	RE-SOURCE_RETIREMENT
GPU-RAS	0x0017	*LTC_FATAL_ERROR*	95	MEM-ORY_INTEGRITY_ERROR
GPU-RAS	0x0018	*LRC_FATAL_ERROR*	48, 172, 95	MEM-ORY_INTEGRITY_ERROR
GPU-RAS	0x0019	*DRAM_ERROR*	48, 171	MEM-ORY_INTEGRITY_ERROR
GPU-RAS	0x001A	*DRAM_SIDEBAND_ERROR*	48, 171	MEM-ORY_INTEGRITY_ERROR
GPU-RAS	0x001B	*DRAM_CBC_FATAL_ERROR*	48, 171, 95	MEM-ORY_INTEGRITY_ERROR
GPU-RAS	0x001C	*DRAM_CBC_NON_FATAL_ERROR*	48, 171, 94	MEM-ORY_INTEGRITY_ERROR
GPU-RAS	0x001D	*LTC_ERROR*	48, 172	MEM-ORY_INTEGRITY_ERROR
GPU-RAS	0x001E	*LTC_UNSUPPORTED_CLIENT_POISON_UNCON*	95	MEM-ORY_INTEGRITY_ERROR
GPU-RAS	0x001F	*LTC_UNSUPPORTED_CLIENT_POISON_CONTAI*	94	MEM-ORY_INTEGRITY_ERROR
GPU-RAS	0x0020	*PCIE_INGRESS_POISON_CONTAINED_ERROR*	94	INTERCON-NECT_ERROR
GPU-RAS	0x0021	*PCIE_INGRESS_POISON_UNCONTAINED_ERRO*	95	INTERCON-NECT_ERROR
GPU-RAS	0x0022	*PCIE_EGRESS_POISON_ERROR*	95	INTERCON-NECT_ERROR
GPU-RAS	0x0023	*FBHUB_FATAL_POISON_ERROR*	95	MEM-ORY_INTEGRITY_ERROR

continues on next page

Table 2 – continued from previous page

Module	Code	Event	Legacy Xid	Category
GPU-RAS	0x0024	*FBHUB_NON_FATAL_POISON_ERROR*	95	MEM-ORY_INTEGRITY_ERROR
GPU-RAS	0x0025	*MMU_POISON_ERROR*	94	MEM-ORY_INTEGRITY_ERROR
GPU-RAS	0x0026	*GSP_POISON_ERROR*	94	MEM-ORY_INTEGRITY_ERROR
GPU-RAS	0x0027	*COPY_ENGINE_POISON_CONTAINED_ERROR*	94	MEM-ORY_INTEGRITY_ERROR
GPU-RAS	0x0028	*COPY_ENGINE_POISON_UNCONTAINED_ERRO*	95	MEM-ORY_INTEGRITY_ERROR
GPU-RAS	0x0029	*PMU_POISON_ERROR*	94	MEM-ORY_INTEGRITY_ERROR
GPU-RAS	0x002A	*SEC2_POISON_ERROR*	94	MEM-ORY_INTEGRITY_ERROR
GPU-RAS	0x002B	*FBFALCON_POISON_ERROR*	94	MEM-ORY_INTEGRITY_ERROR
GPU-RAS	0x002C	*VIDEO_ENGINE_POISON_ERROR*	94	MEM-ORY_INTEGRITY_ERROR
GPU-RAS	0x002D	*SM_POISON_ERROR*	94	MEM-ORY_INTEGRITY_ERROR
GPU-RAS	0x002E	*CONTEXT_SWITCH_POISON_ERROR*	94	MEM-ORY_INTEGRITY_ERROR
GPU-RAS	0x002F	*GCC_POISON_ERROR*	94	MEM-ORY_INTEGRITY_ERROR
GPU-RAS	0x0030	*PCIE_UNCORRECTABLE_ERROR*	48, 172	INTERCON-NECT_ERROR
GPU-RAS	0x0031	*COPY_ENGINE_UNCORRECTABLE_ERROR*	48, 172	MEM-ORY_INTEGRITY_ERROR
GPU-RAS	0x0032	*MMU_UNCORRECTABLE_ERROR*	48, 172	MEM-ORY_INTEGRITY_ERROR
GPU-RAS	0x0033	*FECS_UNCORRECTABLE_ERROR*	48, 172	MEM-ORY_INTEGRITY_ERROR
GPU-RAS	0x0034	*SM_CONTAINED_UNCORRECTABLE_ERROR*	48, 172	MEM-ORY_INTEGRITY_ERROR
GPU-RAS	0x0035	*GCC_UNCORRECTABLE_ERROR*	48, 172	MEM-ORY_INTEGRITY_ERROR
GPU-RAS	0x0036	*CONTEXT_SWITCH_UNCORRECTABLE_ERROR*	48, 172	MEM-ORY_INTEGRITY_ERROR

continues on next page

Table 2 – continued from previous page

Module	Code	Event	Legacy Xid	Category
GPU-RAS	0x0037	*GPCMMU_UNCORRECTABLE_ERROR*	48, 172	MEMORY_INTEGRITY_ERROR
GPU-RAS	0x0038	*GSP_UNCORRECTABLE_ERROR*	48, 172	MEMORY_INTEGRITY_ERROR
GPU-RAS	0x0039	*LPSEC_UNCORRECTABLE_ERROR*	48, 172	MEMORY_INTEGRITY_ERROR
GPU-RAS	0x003A	*NVLINK_UNCORRECTABLE_ERROR*	48, 172	INTERCONNECT_ERROR
GPU-RAS	0x003B	*PMU_UNCORRECTABLE_ERROR*	48, 172	MEMORY_INTEGRITY_ERROR
GPU-RAS	0x003C	*SEC2_UNCORRECTABLE_ERROR*	48, 172	MEMORY_INTEGRITY_ERROR
GPU-RAS	0x003D	*SPARING_EVENT*	185	STATE_CHANGED
GPU-RC	0x0001	*CHANNEL_RESET*	Varies	UNCLASSIFIED_ERROR

3.5.2. Event Details

Entries are grouped by source module signature, then ordered by category value and module event code. NVLink secure events follow the other NVLink events.

3.5.2.1 GPU

3.5.2.1.1 DRIVER_INIT

Code
0x0001

Category
INITIALIZATION

Severity
INFORMATIONAL

Log level
NOTICE

Scope
DEVICE

Legacy Xid
None

GPU driver initialization completed for the device. Carries driver and device metadata.

3.5.2.2 GPU-BUS

3.5.2.2.1 C2C_CONTAINMENT

Code
0x0002

Category
INTERCONNECT_ERROR

Severity
FATAL or RECOVERABLE

Log level
ERROR

Scope
DEVICE

Legacy Xid
179

C2C containment was triggered on the chip-to-chip link to the host CPU. Carries the containment code.

3.5.2.2.2 GPU_FALLEN_OFF_BUS

Code
0x0003

Category
INTERCONNECT_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
79

The GPU stopped responding to register reads and is no longer accessible on the bus. Only emitted on Unix platforms.

3.5.2.3 GPU-FSP

3.5.2.3.1 BOOT_TIMEOUT

Code
0x0001

Category
TIMEOUT

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
143

The driver timed out polling for FSP boot completion during GPU initialization.

3.5.2.3.2 FUSE_ERROR

Code
0x0002

Category
FIRMWARE_FAULT

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
143

The FSP fuse error check failed. Carries the fuse status.

3.5.2.4 GPU-GSP

3.5.2.4.1 RPC_TIMEOUT

Code
0x0001

Category
TIMEOUT

Severity
FATAL or RECOVERABLE

Log level
ERROR or WARNING

Scope
DEVICE

Legacy Xid
119

The driver timed out waiting for an RPC response from GSP. Carries the RPC function, name, sequence number, and wait duration.

3.5.2.4.2 LIBOS_HEARTBEAT_TIMEOUT

Code
0x0002

Category
TIMEOUT

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
120

The LibOS heartbeat from the GSP microcontroller stopped. Emitted only when GSP timeout classification is enabled; disabled by default.

3.5.2.4.3 GSP_RM_HEARTBEAT_TIMEOUT

Code
0x0003

Category
TIMEOUT

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
120

The GSP-RM heartbeat stopped. Emitted only when GSP timeout classification is enabled; disabled by default.

3.5.2.4.4 POISON

Code
0x0005

Category
MEMORY_INTEGRITY_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
140

GSP consumed poisoned data.

3.5.2.4.5 FIRMWARE_FAULT

Code
0x0004

Category
FIRMWARE_FAULT

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
120

GSP firmware reported a fault.

3.5.2.5 GPU-MEMSYS

3.5.2.5.1 MAINT_OP_ERROR

Code
0x0001

Category
MEMORY_INTEGRITY_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
175

A GPU memory subsystem maintenance operation timed out. Carries the operation type.

3.5.2.6 GPU-NVLINK

The SAW, RLW, TLW, TREX, NVLPW, and NETIR events carry the link ID, interrupt information, error status, and debug registers. Their severity and log level distinguish warnings, contained errors, and fatal errors.

3.5.2.6.1 ALI_TRAINING_FAILURE

Code
0x0001

Category
INTERCONNECT_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
136

NVLink ALI link training failed. Carries the affected link IDs and debug registers.

3.5.2.6.2 SW_LINK_DOWN

Code
0x0004

Category
INTERCONNECT_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
155

An NVLink went down while it was active. Carries the link ID.

3.5.2.6.3 FATAL_ERROR_NVL3

Code
0x0005

Category
INTERCONNECT_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
74

A fatal NVLink error occurred. Carries the link ID and debug registers.

3.5.2.6.4 PRIV_ERROR

Code
0x0006

Category
INTERCONNECT_ERROR

Severity
INFORMATIONAL

Log level
WARNING

Scope
DEVICE

Legacy Xid
137

An NVLink privileged-register access failed. Carries the link ID.

3.5.2.6.5 SAW_ERROR_GPU_FATAL

Code
0x0007

Category
INTERCONNECT_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
144

A fatal NVLink SAW error requires a GPU reset.

3.5.2.6.6 SAW_ERROR_NON_FATAL_WARNING

Code
0x0008

Category
INTERCONNECT_ERROR

Severity
INFORMATIONAL

Log level
WARNING

Scope
DEVICE

Legacy Xid
144

NVLink SAW reported a nonfatal warning.

3.5.2.6.7 SAW_ERROR_NON_FATAL_CONTAINED

Code
0x0009

Category
INTERCONNECT_ERROR

Severity
RECOVERABLE

Log level
ERROR

Scope
DEVICE

Legacy Xid
144

A contained, recoverable NVLink SAW error occurred.

3.5.2.6.8 RLW_ERROR_GPU_FATAL

Code
0x000A

Category
INTERCONNECT_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
145

A fatal NVLink RLW error requires a GPU reset.

3.5.2.6.9 RLW_ERROR_NON_FATAL_WARNING

Code
0x000B

Category
INTERCONNECT_ERROR

Severity
INFORMATIONAL

Log level
WARNING

Scope
DEVICE

Legacy Xid
145

NVLink RLW reported a nonfatal warning.

3.5.2.6.10 RLW_ERROR_APPLICATION_FATAL

Code
0x000C

Category
INTERCONNECT_ERROR

Severity
RECOVERABLE

Log level
ERROR

Scope
DEVICE

Legacy Xid
145

An NVLink RLW error affected an application while the GPU remains recoverable.

3.5.2.6.11 TLW_ERROR_GPU_FATAL

Code
0x000D

Category
INTERCONNECT_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
146

A fatal NVLink TLW error requires a GPU reset.

3.5.2.6.12 TLW_ERROR_NON_FATAL_WARNING

Code
0x000E

Category
INTERCONNECT_ERROR

Severity
INFORMATIONAL

Log level
WARNING

Scope
DEVICE

Legacy Xid
146

NVLink TLW reported a nonfatal warning.

3.5.2.6.13 TLW_ERROR_NON_FATAL_CONTAINED

Code
0x000F

Category
INTERCONNECT_ERROR

Severity
RECOVERABLE

Log level
ERROR

Scope
DEVICE

Legacy Xid
146

A contained, recoverable NVLink TLW error occurred.

3.5.2.6.14 TREX_ERROR_NON_FATAL_WARNING

Code
0x0010

Category
INTERCONNECT_ERROR

Severity
INFORMATIONAL

Log level
WARNING

Scope
DEVICE

Legacy Xid
147

NVLink TREX reported a nonfatal warning.

3.5.2.6.15 NVLPW_ERROR_NON_FATAL_WARNING

Code
0x0011

Category
INTERCONNECT_ERROR

Severity
INFORMATIONAL

Log level
WARNING

Scope
DEVICE

Legacy Xid
148

NVLink NVLPW reported a nonfatal warning.

3.5.2.6.16 NETIR_ERROR_NON_FATAL_WARNING

Code
0x0012

Category
INTERCONNECT_ERROR

Severity
INFORMATIONAL

Log level
WARNING

Scope
DEVICE

Legacy Xid
149

NVLink NETIR reported a nonfatal warning.

3.5.2.6.17 NETIR_ERROR_GPU_FATAL

Code
0x0013

Category
INTERCONNECT_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
149

A fatal NVLink NETIR error requires a GPU reset.

3.5.2.6.18 NETIR_ERROR_LINK_FATAL

Code
0x0014

Category
INTERCONNECT_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
149

A fatal NVLink NETIR error affected a link.

3.5.2.6.19 MSE_DEGRADED

Code
0x0002

Category
FIRMWARE_FAULT

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
150

The NVLink MSE transport entered a degraded state. The driver stops MSE communication until it is reloaded.

3.5.2.6.20 MSE_WATCHDOG_TIMEOUT

Code
`0x0003`

Category
`FIRMWARE_FAULT`

Severity
`FATAL` or `RECOVERABLE`

Log level
`ERROR`

Scope
`DEVICE`

Legacy Xid
`150`

The NVLink MSE watchdog expired. Carries the error status and debug registers.

The following secure events carry the secure interrupt status, link ID, error-log enable bits, and error status.

3.5.2.6.21 IR2SAW_INTERRUPT

Code
`0x0001`

Category
`INTERCONNECT_ERROR`

Severity
`FATAL`

Log level
`ERROR`

Scope
`DEVICE`

Legacy Xid
`170`

Context type
`0xA501`

A fatal NVLink IR2SAW secure interrupt occurred.

3.5.2.6.22 CROSS_CONTAIN

Code
`0x0002`

Category
`INTERCONNECT_ERROR`

Severity
`FATAL`

Log level
ERROR

Scope
DEVICE

Legacy Xid
170

Context type
0xA502

A fatal NVLink cross-containment error occurred.

3.5.2.6.23 ENCRYPT_ERROR

Code
0x0003

Category
INTERCONNECT_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
170

Context type
0xA503

An NVLink encryption error occurred.

3.5.2.6.24 REMOTE_HW_ERR

Code
0x0004

Category
INTERCONNECT_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
170

Context type
0xA504

An NVLink remote hardware error occurred.

3.5.2.6.25 LOCAL_STO_ERR

Code
0x0005

Category
INTERCONNECT_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
170

Context type
0xA505

An NVLink local STO error occurred.

3.5.2.6.26 LOCAL_PRIV_ERR

Code
0x0006

Category
INTERCONNECT_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
170

Context type
0xA506

An NVLink local privileged-register error occurred.

3.5.2.6.27 REMOTE_PRIV_ERR

Code
0x0007

Category
INTERCONNECT_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
170

Context type
0xA507

An NVLink remote privileged-register error occurred.

3.5.2.6.28 DMEM_PARITY_ERR

Code
0x0008

Category
INTERCONNECT_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
170

Context type
0xA508

An NVLink data-memory parity error occurred.

3.5.2.6.29 IMEM_PARITY_ERR

Code
0x0009

Category
INTERCONNECT_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
170

Context type
0xA509

An NVLink instruction-memory parity error occurred.

3.5.2.6.30 MPU_PARITY_ERR

Code
0x000A

Category
INTERCONNECT_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
170

Context type
0xA50A

An NVLink MPU parity error occurred.

3.5.2.6.31 REG_PARITY_ERR

Code
0x000B

Category
INTERCONNECT_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
170

Context type
0xA50B

An NVLink register parity error occurred.

3.5.2.6.32 TRAFFIC_ERR

Code
0x000C

Category
INTERCONNECT_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
170

Context type
0xA50C

An NVLink secure-traffic error occurred.

3.5.2.6.33 ICACHE_PARITY_ERR

Code
0x000D

Category
INTERCONNECT_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
170

Context type
0xA50D

An NVLink instruction-cache parity error occurred.

3.5.2.6.34 DCACHE_PARITY_ERR

Code
0x000E

Category
INTERCONNECT_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
170

Context type
0xA50E

An NVLink data-cache parity error occurred.

3.5.2.6.35 IR2SAW_RECOVERABLE

Code
0x000F

Category
INTERCONNECT_ERROR

Severity
RECOVERABLE

Log level
ERROR

Scope
DEVICE

Legacy Xid
170

Context type
0xA501

A recoverable NVLink IR2SAW secure interrupt occurred.

3.5.2.6.36 CROSS_CONTAIN_RECOVERABLE

Code
0x0010

Category
INTERCONNECT_ERROR

Severity
RECOVERABLE

Log level
ERROR

Scope
DEVICE

Legacy Xid
170

Context type
0xA502

A recoverable NVLink cross-containment error occurred.

3.5.2.6.37 LOCAL_STO_ERR_RECOVERABLE

Code
0x0011

Category
INTERCONNECT_ERROR

Severity
RECOVERABLE

Log level
ERROR

Scope
DEVICE

Legacy Xid
170

Context type
0xA505

A recoverable NVLink local STO error occurred.

3.5.2.7 GPU-PCIE

3.5.2.7.1 FATAL_TIMEOUT

Code
0x0001

Category
INTERCONNECT_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
167

A fatal timeout occurred in the PCIe endpoint layer. Carries the error type and register value.

3.5.2.7.2 CORRECTABLE_ERROR_FAULT

Code
0x0002

Category
INTERCONNECT_ERROR

Severity
CORRECTED

Log level
NOTICE

Scope
DEVICE

Legacy Xid
176

The PCIe correctable error rate exceeded a threshold. Carries the error-rate tuple identifiers. Native CPER output omits the event-specific and legacy Xid contexts.

3.5.2.7.3 RESET_FAILURE

Code
0x0003

Category
INTERCONNECT_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
182

A PCIe reset failed. Carries the reset status and recovery action.

3.5.2.8 GPU-PSHC

3.5.2.8.1 PSHC_REENGAGED

Code
0x0001

Category
STATE_CHANGED

Severity
INFORMATIONAL

Log level
DIAGNOSTIC

Scope
DEVICE

Legacy Xid
162

The power-smoothing controller reengaged.

3.5.2.8.2 PSHC_DISENGAGED

Code
0x0002

Category
STATE_CHANGED

Severity
INFORMATIONAL

Log level
DIAGNOSTIC

Scope
DEVICE

Legacy Xid
163

The power-smoothing controller disengaged because of a slowdown, timeout, or exhausted lifetime. Carries the reason and, for a slowdown, the slowdown mask.

3.5.2.8.3 PSHC_LOW_LIFETIME

Code
0x0003

Category
RESOURCE_THRESHOLD

Severity
INFORMATIONAL

Log level
WARNING

Scope
DEVICE

Legacy Xid
164

The power-smoothing controller lifetime fell below a threshold. Carries the threshold percentage.

3.5.2.8.4 PSHC_ZERO_LIFETIME

Code
0x0004

Category
RESOURCE_THRESHOLD

Severity
INFORMATIONAL

Log level
WARNING

Scope
DEVICE

Legacy Xid
165

The power-smoothing controller lifetime reached zero. The controller will be disabled.

3.5.2.9 GPU-RAS

3.5.2.9.1 ECC_RESIDUAL_UNCORRECTABLE_ERROR

Code
0x0013

Category
MEMORY_INTEGRITY_ERROR

Severity
FATAL or RECOVERABLE

Log level
ERROR

Scope
DEVICE

Legacy Xid
140

An uncorrectable ECC error was detected outside the normal handling path.

3.5.2.9.2 DRAM_ECC_INTR_STORM

Code
0x0014

Category
MEMORY_INTEGRITY_ERROR

Severity
CORRECTED

Log level
WARNING

Scope
DEVICE

Legacy Xid
92

Correctable ECC error interrupts were disabled on a framebuffer partition because of a high error rate.

3.5.2.9.3 SM_ECC_INTR_STORM

Code
0x0015

Category
MEMORY_INTEGRITY_ERROR

Severity
CORRECTED

Log level
WARNING

Scope
DEVICE

Legacy Xid
92

An SM correctable ECC interrupt storm was detected.

3.5.2.9.4 LTC_FATAL_ERROR

Code
0x0017

Category
MEMORY_INTEGRITY_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
95

An uncontained L2 data, tag, or compression-tag error requires a GPU reset.

3.5.2.9.5 LRC_FATAL_ERROR

Code
0x0018

Category
MEMORY_INTEGRITY_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
48, 172, 95

An uncontained LRC SRAM error requires a GPU reset.

3.5.2.9.6 DRAM_ERROR

Code
0x0019

Category
MEMORY_INTEGRITY_ERROR

Severity
RECOVERABLE

Log level
ERROR

Scope
DEVICE

Legacy Xid
48, 171

An uncorrectable DRAM error occurred.

3.5.2.9.7 DRAM_SIDEBAND_ERROR

Code
0x001A

Category
MEMORY_INTEGRITY_ERROR

Severity
RECOVERABLE

Log level
ERROR

Scope
DEVICE

Legacy Xid
48, 171

An uncorrectable DRAM sideband error occurred.

3.5.2.9.8 DRAM_CBC_FATAL_ERROR

Code
0x001B

Category
MEMORY_INTEGRITY_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
48, 171, 95

An uncontained DRAM compression-tag error requires a GPU reset.

3.5.2.9.9 DRAM_CBC_NON_FATAL_ERROR

Code
0x001C

Category
MEMORY_INTEGRITY_ERROR

Severity
RECOVERABLE

Log level
ERROR

Scope
DEVICE

Legacy Xid
48, 171, 94

A contained DRAM compression-tag error occurred.

3.5.2.9.10 LTC_ERROR

Code
0x001D

Category
MEMORY_INTEGRITY_ERROR

Severity
RECOVERABLE

Log level
ERROR

Scope
DEVICE

Legacy Xid
48, 172

An uncorrectable L2 SRAM error occurred.

3.5.2.9.11 LTC_UNSUPPORTED_CLIENT_POISON_UNCONTAINED_ERROR

Code
0x001E

Category
MEMORY_INTEGRITY_ERROR

Severity
RECOVERABLE

Log level
ERROR

Scope
DEVICE

Legacy Xid
95

An L2 client that does not support poison reported an uncontained error.

3.5.2.9.12 LTC_UNSUPPORTED_CLIENT_POISON_CONTAINED_ERROR

Code
0x001F

Category
MEMORY_INTEGRITY_ERROR

Severity
RECOVERABLE

Log level
ERROR

Scope
DEVICE

Legacy Xid
94

An L2 client that does not support poison reported a contained error.

3.5.2.9.13 FBHUB_FATAL_POISON_ERROR

Code
0x0023

Category
MEMORY_INTEGRITY_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
95

An uncontained framebuffer hub poison error requires a GPU reset.

3.5.2.9.14 FBHUB_NON_FATAL_POISON_ERROR

Code
0x0024

Category
MEMORY_INTEGRITY_ERROR

Severity
RECOVERABLE

Log level
ERROR

Scope
DEVICE

Legacy Xid
95

A framebuffer hub poison error occurred without requiring an immediate GPU reset.

3.5.2.9.15 MMU_POISON_ERROR

Code
0x0025

Category
MEMORY_INTEGRITY_ERROR

Severity
RECOVERABLE

Log level
ERROR

Scope
ENGINE

Legacy Xid
94

The MMU reported a contained poison error.

3.5.2.9.16 GSP_POISON_ERROR

Code
0x0026

Category
MEMORY_INTEGRITY_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
94

GSP reported a contained poison error that requires a GPU reset.

3.5.2.9.17 COPY_ENGINE_POISON_CONTAINED_ERROR

Code
0x0027

Category
MEMORY_INTEGRITY_ERROR

Severity
RECOVERABLE

Log level
ERROR

Scope
ENGINE

Legacy Xid
94

A copy-engine user channel reported a contained poison error.

3.5.2.9.18 COPY_ENGINE_POISON_UNCONTAINED_ERROR

Code
0x0028

Category
MEMORY_INTEGRITY_ERROR

Severity
FATAL

Log level
ERROR

Scope
ENGINE

Legacy Xid
95

A copy-engine kernel channel reported an uncontained poison error requiring a GPU reset.

3.5.2.9.19 PMU_POISON_ERROR

Code
0x0029

Category
MEMORY_INTEGRITY_ERROR

Severity
FATAL

Log level
ERROR

Scope
ENGINE

Legacy Xid
94

PMU reported a contained poison error that requires a GPU reset.

3.5.2.9.20 SEC2_POISON_ERROR

Code
0x002A

Category
MEMORY_INTEGRITY_ERROR

Severity
FATAL

Log level
ERROR

Scope
ENGINE

Legacy Xid
94

SEC2 reported a contained poison error that requires a GPU reset.

3.5.2.9.21 FBFALCON_POISON_ERROR

Code
0x002B

Category
MEMORY_INTEGRITY_ERROR

Severity
FATAL

Log level
ERROR

Scope
ENGINE

Legacy Xid
94

The framebuffer Falcon reported a contained poison error that requires a GPU reset.

3.5.2.9.22 VIDEO_ENGINE_POISON_ERROR

Code
0x002C

Category
MEMORY_INTEGRITY_ERROR

Severity
RECOVERABLE

Log level
ERROR

Scope
ENGINE

Legacy Xid
94

A video engine reported a contained poison error.

3.5.2.9.23 SM_POISON_ERROR

Code
0x002D

Category
MEMORY_INTEGRITY_ERROR

Severity
RECOVERABLE

Log level
ERROR

Scope
ENGINE

Legacy Xid
94

An SM reported a contained poison error.

3.5.2.9.24 CONTEXT_SWITCH_POISON_ERROR

Code
0x002E

Category
MEMORY_INTEGRITY_ERROR

Severity
RECOVERABLE

Log level
ERROR

Scope
ENGINE

Legacy Xid
94

The context-switch engine reported a contained poison error.

3.5.2.9.25 GCC_POISON_ERROR

Code
0x002F

Category
MEMORY_INTEGRITY_ERROR

Severity
RECOVERABLE

Log level
ERROR

Scope
ENGINE

Legacy Xid
94

GCC reported a contained poison error.

3.5.2.9.26 COPY_ENGINE_UNCORRECTABLE_ERROR

Code
0x0031

Category
MEMORY_INTEGRITY_ERROR

Severity
RECOVERABLE

Log level
ERROR

Scope
ENGINE

Legacy Xid
48, 172

An uncorrectable SRAM error occurred in a copy engine.

3.5.2.9.27 MMU_UNCORRECTABLE_ERROR

Code
0x0032

Category
MEMORY_INTEGRITY_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
48, 172

An uncorrectable SRAM error occurred in the MMU.

3.5.2.9.28 FECS_UNCORRECTABLE_ERROR

Code
0x0033

Category
MEMORY_INTEGRITY_ERROR

Severity
RECOVERABLE

Log level
ERROR

Scope
ENGINE

Legacy Xid
48, 172

An uncorrectable SRAM error occurred in FECS.

3.5.2.9.29 SM_CONTAINED_UNCORRECTABLE_ERROR

Code
`0x0034`

Category
`MEMORY_INTEGRITY_ERROR`

Severity
`RECOVERABLE`

Log level
`ERROR`

Scope
`ENGINE`

Legacy Xid
48, 172

A contained uncorrectable SRAM error occurred in an SM.

3.5.2.9.30 GCC_UNCORRECTABLE_ERROR

Code
`0x0035`

Category
`MEMORY_INTEGRITY_ERROR`

Severity
`RECOVERABLE`

Log level
`ERROR`

Scope
`ENGINE`

Legacy Xid
48, 172

An uncorrectable SRAM error occurred in GCC.

3.5.2.9.31 CONTEXT_SWITCH_UNCORRECTABLE_ERROR

Code
`0x0036`

Category
`MEMORY_INTEGRITY_ERROR`

Severity
`RECOVERABLE`

Log level
ERROR

Scope
ENGINE

Legacy Xid
48, 172

An uncorrectable SRAM error occurred in GPCCS.

3.5.2.9.32 GPCMMU_UNCORRECTABLE_ERROR

Code
0x0037

Category
MEMORY_INTEGRITY_ERROR

Severity
FATAL

Log level
ERROR

Scope
ENGINE

Legacy Xid
48, 172

An uncorrectable SRAM error occurred in GPCMMU.

3.5.2.9.33 GSP_UNCORRECTABLE_ERROR

Code
0x0038

Category
MEMORY_INTEGRITY_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
48, 172

An uncorrectable SRAM error occurred in GSP-RISC-V.

3.5.2.9.34 LPSEC_UNCORRECTABLE_ERROR

Code
0x0039

Category
MEMORY_INTEGRITY_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
48, 172

An uncorrectable SRAM error occurred in LPSEC.

3.5.2.9.35 PMU_UNCORRECTABLE_ERROR

Code
0x003B

Category
MEMORY_INTEGRITY_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
48, 172

An uncorrectable SRAM error occurred in the PMU Falcon or RISC-V core.

3.5.2.9.36 SEC2_UNCORRECTABLE_ERROR

Code
0x003C

Category
MEMORY_INTEGRITY_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
48, 172

An uncorrectable SRAM error occurred in SEC2-RISC-V.

3.5.2.9.37 PCIE_INGRESS_POISON_CONTAINED_ERROR

Code
0x0020

Category
INTERCONNECT_ERROR

Severity
RECOVERABLE

Log level
ERROR

Scope
DEVICE

Legacy Xid
94

A contained PCIe ingress poison error occurred.

3.5.2.9.38 PCIE_INGRESS_POISON_UNCONTAINED_ERROR

Code
0x0021

Category
INTERCONNECT_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
95

An uncontained PCIe ingress poison error requires a GPU reset.

3.5.2.9.39 PCIE_EGRESS_POISON_ERROR

Code
0x0022

Category
INTERCONNECT_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
95

An uncontained PCIe egress poison error requires a GPU reset.

3.5.2.9.40 PCIE_UNCORRECTABLE_ERROR

Code
0x0030

Category
INTERCONNECT_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
48, 172

An uncorrectable SRAM error occurred in PCIe.

3.5.2.9.41 NVLINK_UNCORRECTABLE_ERROR

Code
0x003A

Category
INTERCONNECT_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
48, 172

An uncorrectable SRAM error occurred in NVLink.

3.5.2.9.42 ROW_REMAPPING_FAILURE_TABLE_FULL

Code
0x0002

Category
RESOURCE_EXHAUSTED

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
64

Row remapping failed because the row remapping table is full.

3.5.2.9.43 ROW_REMAPPING_FAILURE_BANK_FULL

Code
0x0003

Category
RESOURCE_EXHAUSTED

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
64

Row remapping failed because all reserved rows for the bank are already remapped.

3.5.2.9.44 ROW_REMAPPING_FAILURE_RESERVED_ROW

Code
0x0005

Category
RESOURCE_EXHAUSTED

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
64

Row remapping failed because the target is a reserved row.

3.5.2.9.45 DRAM_RETIREMENT_FAILURE_INFOROM_FULL

Code
0x0008

Category
RESOURCE_EXHAUSTED

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
64

DRAM retirement failed because the InfoROM object is full.

3.5.2.9.46 DRAM_RETIREMENT_FAILURE_HW_LIMIT

Code
0x0009

Category
RESOURCE_EXHAUSTED

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
64

DRAM retirement failed because a hardware limit was reached.

3.5.2.9.47 DRAM_RETIREMENT_FAILURE_NO_SPARE

Code
0x000A

Category
RESOURCE_EXHAUSTED

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
64

DRAM retirement failed because no spare remains for retirement.

3.5.2.9.48 LTS_REPAIR_FAILURE

Code
0x000C

Category
RESOURCE_EXHAUSTED

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
161

L2 slice repair failed because no spare L2 slices remain.

3.5.2.9.49 MEMORY_CHANNEL_REPAIR_FAILURE

Code
0x000E

Category
RESOURCE_EXHAUSTED

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
161

Memory channel repair failed because no spare channels remain.

3.5.2.9.50 SPARING_REPAIR_NOT_POSSIBLE

Code
0x0011

Category
RESOURCE_EXHAUSTED

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
157

A repair using spare resources is not possible.

3.5.2.9.51 ROW_REMAPPING_FAILURE_INTERNAL_ERROR

Code
0x0004

Category
UNCLASSIFIED_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
64

Row remapping failed due to an internal driver error.

3.5.2.9.52 ROW_REMAPPING_FAILURE_PAGE_OFFLINE_FAILURE

Code
0x0006

Category
UNCLASSIFIED_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
64

The row is already pending remapping. The remap occurs on the next GPU reset.

3.5.2.9.53 DRAM_RETIREMENT_FAILURE_INTERNAL_ERROR

Code
0x0007

Category
UNCLASSIFIED_ERROR

Severity
FATAL

Log level
ERROR

Scope
DEVICE

Legacy Xid
64

DRAM retirement failed due to an internal driver error.

3.5.2.9.54 SPARING_EVENT

Code
0x003D

Category
STATE_CHANGED

Severity
RECOVERABLE

Log level
NOTICE

Scope
DEVICE

Legacy Xid
185

A sparing operation was reported.

3.5.2.9.55 ROW_REMAPPING_PENDING

Code
0x0001

Category
RESOURCE_RETIREMENT

Severity
RECOVERABLE

Log level
NOTICE

Scope
DEVICE

Legacy Xid
63

A new DRAM row was marked for remapping. The remap activates on the next GPU reset.

3.5.2.9.56 LTS_REPAIR_PENDING

Code
0x000B

Category
RESOURCE_RETIREMENT

Severity
RECOVERABLE

Log level
NOTICE

Scope
DEVICE

Legacy Xid
160

An L2 slice and its pair were marked for repair. The Xid 160 message gives the location and the action needed to activate the repair.

3.5.2.9.57 MEMORY_CHANNEL_REPAIR_PENDING

Code
0x000D

Category
RESOURCE_RETIREMENT

Severity
RECOVERABLE

Log level
NOTICE

Scope
DEVICE

Legacy Xid
160

A memory channel and its pair were marked for repair. The Xid 160 message gives the location and the action needed to activate the repair.

3.5.2.9.58 SPARING_REPAIR_PENDING

Code
0x000F

Category
RESOURCE_RETIREMENT

Severity
RECOVERABLE

Log level
NOTICE

Scope
DEVICE

Legacy Xid
156

A repair using spare resources is pending.

3.5.2.9.59 BANK_REMAPPING_PENDING

Code
0x0016

Category
RESOURCE_RETIREMENT

Severity
RECOVERABLE

Log level
NOTICE

Scope
DEVICE

Legacy Xid
177

A new DRAM bank was marked for remapping. The remap activates on the next GPU reset.

3.5.2.10 GPU-RC

3.5.2.10.1 CHANNEL_RESET

Code
0x0001

Category
UNCLASSIFIED_ERROR

Severity
RECOVERABLE

Log level
ERROR

Scope
ENGINE

Legacy Xid
Varies

Robust Channel recovery reset a channel or engine. The Legacy Xid context identifies the originating Xid. Additional recovery details are carried in an opaque context.

Chapter 4. Release Notes

GPU Operational Events remain a preview feature. This page describes changes in R620 from R615. The public NVML event APIs, category values, severity values, and CPER section layout are unchanged. Consumers must still update their event and context decoders for the changes below.

4.1. New Event Coverage

Module	Added Codes	Conditions
GPU-BUS	0x0003	GPU fallen off the bus (Xid 79).
GPU-NVLINK	0x0005–0x0014	Fatal link errors, privilege errors, and SAW, RLW, TLW, TREX, NVLPW, and NETIR errors.
GPU-NVLINK (secure)	0x0001–0x0011	Secure interrupts, containment, encryption, privilege, traffic, and parity errors.
GPU-PCIE	0x0001–0x0003	Fatal timeouts, correctable errors, and reset failures.
GPU-PSHC	0x0001–0x0004	Power-smoothing controller engagement, disengagement, and low or exhausted lifetime.
GPU-RAS	0x0017–0x003D	LTC/LRC and DRAM errors, contained and uncontained poison, SRAM ECC errors, and sparing state changes. SPARING_EVENT (0x003D) reports Xid 185.

The *GPU Operational Event Catalog* lists each event's category, severity, scope, log level, and legacy Xid. Event availability depends on the GPU and the reporting path.

4.2. New Reporting Capabilities

- The Linux open kernel module exposes GPU Operational Events through `nvidia` tracepoints, selected by log level. See *Linux Tracepoints*.
- CPER record flags now reflect the event group's recovered, previous-error, and simulated attributes. Imported CPER timestamps are marked imprecise.
- Legacy Xid messages for simulated events include `[SIMULATED]`.

Chapter 5. Notices

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. NVIDIA Corporation (“NVIDIA”) makes no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice.

Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer (“Terms of Sale”). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

NVIDIA products are not designed, authorized, or warranted to be suitable for use in medical, military, aircraft, space, or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death, or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer’s own risk.

NVIDIA makes no representation or warranty that products based on this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer’s sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer’s product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this document. NVIDIA accepts no liability related to any default, damage, costs, or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this document or (ii) customer product designs.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this document. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA.

Reproduction of information in this document is permissible only if approved in advance by NVIDIA in writing, reproduced without alteration and in full compliance with all applicable export laws and regulations, and accompanied by all associated conditions, limitations, and notices.

THIS DOCUMENT AND ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, “MATERIALS”) ARE BEING PROVIDED “AS IS.” NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA’s aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms of Sale for the product.

5.1. Trademarks

NVIDIA and the NVIDIA logo are trademarks or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

©2026, NVIDIA Corporation & affiliates. All rights reserved