



# Baseboard Management Controller Messages

The Baseboard Management Controller (BMC) provides the interface to the System Event Log (SEL). The SEL can be accessed from the system side as well as from other external interfaces. The BMC uses a message handler to route messages between the different interfaces. It also monitors and manages the system board, including temperatures and voltages.

The following sections are included:

- [SEL Device, page 7-1](#)
- [Sensor Data Record Device, page 7-2](#)
- [Event Receiver Device, page 7-3](#)
- [BMC Commands, page 7-3](#)
- [SEL Record Examples, page 7-8](#)

## SEL Device

The SEL is a nonvolatile repository for system events. The SEL device is separate from the event receiver device and accepts commands to manage the contents.

This section includes the following topics:

- [SEL Event Record Format, page 7-1](#)
- [Sensor Initialization Agent, page 7-2](#)

## SEL Event Record Format

The SEL messages are logged as a 16 byte string that contains the information about the change that triggered the message.

- Byte 1 and 2 is the record ID.
- Byte 3 is the record type.
- Bytes 4, 5, 6, and 7 is the timestamp
- Bytes 8 and 9 is the generator ID.
- Byte 10 is the version of the event message format.
- Byte 11 is the sensor type.
- Byte 12 is the sensor number.

- Byte 13 is either the event dir (assertion/deassertion event) or the event type.
- Byte 14, 15, and 16 are links to the event data field contents and determines whether the sensor class is about threshold, discrete, or original equipment manufacturer (OEM) settings.

## Sensor Initialization Agent

The Sensor Initialization Agent is not a logical device, but a collection of functions and services specific to handling SDR information. The Sensor Initialization Agent works directly with the content of SDRs, in particular, with the sensor data records and the device locator records.

The agent uses the SDR information for sensor and IPMB device initialization during system startup. The agent interprets sensor data records and is directed by the *init required* fields to load thresholds to sensors that have the *threshold initialization required* bit set in the SDR records. Other bits in the record direct the agent to enable sensors and devices that come up with sensors, events, or both disabled.

The agent function runs at system power-up and at any system hard resets. We recommend that you run the agent function when the BMC first receives standby power.

In systems that implement power management, the system management software takes additional steps to restore intermediate settings after the system has powered up.

## Sensor Data Record Device

The Sensor Data Record (SDR) device provides the interface to the sensor data records. A set of commands store and retrieve sensor data records. The SDR device provides a set of commands for discovering, configuring, and accessing sensors.

This section includes the following topics:

- [SDR Repository Interface, page 7-2](#)
- [Modal and Nonmodal SDR Repositories, page 7-2](#)

## SDR Repository Interface

The SDR repository holds sensor, device locator, and entity association records for all sensors in the platform management subsystem. The BMC provides this interface to the SDR repository. The sensor data records can be accessed by using SDR commands.

## Modal and Nonmodal SDR Repositories

There are two SDR repository implementations: modal and nonmodal.

A modal SDR repository is only updated when the controller is in SDR repository update mode. SDR information is kept in nonvolatile storage devices. Lengthy write operations during update can be required, which can interfere with other controller operations. For example, the SDR repository can be stored in a flash device that also holds a portion of the management controller code. A modal SDR repository implementation allows the functions associated with that code to be temporarily unavailable during the update process.

A nonmodal SDR repository can be written to at any time. Writing to the SDR does not impact the operation of other commands in the management controller.

# Event Receiver Device

Event messages are special messages sent to management controllers when they detect significant or critical system management events. This includes messages for events such as temperature threshold exceeded, voltage threshold exceeded, power fault, and so on. The device generating an event message notifies the system by sending the message to the event receiver device.

Messages from the event receiver device are directly written into the system event log. The appropriate **Add SEL Entry** command is sent directly to the SEL device.

## BMC Commands

SEL, SDR, and event commands are designed so that the devices that implement those command sets are isolated from the contents of the message. The devices do not interpret the messages. The event receiver device receives and routes event messages. The SEL devices retrieve and store log entries. The SDR devices retrieve and store sensor data records.

This section includes the following topics:

- [SEL Device Commands, page 7-3](#)
- [SDR Repository Device Commands, page 7-5](#)
- [Event Receiver Commands, page 7-7](#)

## SEL Device Commands

These are the available SEL device commands:

- [Get SEL Info, page 7-3](#)
- [Get SEL Allocation Info, page 7-4](#)
- [Reserve SEL, page 7-4](#)
- [Get SEL Entry, page 7-4](#)
- [Add SEL Entry, page 7-4](#)
- [Partial Add SEL Entry, page 7-4](#)
- [Delete SEL Entry, page 7-4](#)
- [Clear SEL, page 7-4](#)
- [Get SEL Time, page 7-4](#)
- [Set SEL Time, page 7-5](#)
- [Get Auxiliary Log Status, page 7-5](#)
- [Set Auxiliary Log Status, page 7-5](#)

### Get SEL Info

This command returns the number of entries in the SEL, the SEL command version, and the timestamp for the most recent entry and delete or clear.

## Get SEL Allocation Info

This command returns the number of possible allocation units, the amount of usable free space (in allocation units), the allocation unit size (in bytes), and the size of the largest contiguous free region (in allocation units). The allocation unit size is the number of bytes in which storage is allocated. For example, if a 16 byte record is to be added, and the SEL has a 32 byte allocation unit size, the record takes up 32 bytes of storage.

## Reserve SEL

This command sets the present owner of the SEL, as identified by the software ID or by the requester slave address from the command. The reservation process provides a limited amount of protection at repository access from the Intelligent Platform Management Interface (IPMB) when records are being deleted or incrementally read.

## Get SEL Entry

This command retrieves entries from the SEL. The record data field in the response returns the 16 bytes of data from the SEL event record.

## Add SEL Entry

This command enables the BIOS to add records to the system event log. Normally, the SEL device and the event receiver service are incorporated into the same management controller. In this case, BIOS or the system SMI handler adds its own events to the SEL by formatting an event message and sending it to the SEL device rather than by using this command.

## Partial Add SEL Entry

This command is a version of the **Add SEL Entry** command. It allows the record to be incrementally added to the SEL. This command must be preceded by a **Reserve SEL** command. The first partial add must be to offset 0000h, and subsequent partial adds must be done sequentially, with no gaps or overlap between the adds.

## Delete SEL Entry

This command deletes the specified entry in the SEL.

## Clear SEL

This command erases the SEL contents. This process can take several seconds, based on the type of storage device. The command also shows the status of the erasure.

## Get SEL Time

This command returns the time from the SEL device, which uses it for event timestamps.

## Set SEL Time

This command initializes the time setting in the SEL device, which uses it for event timestamps.

## Get Auxiliary Log Status

This command allows remote software to know whether new information has been added to machine check architecture (MCA) log. The MCA log is a storage area that can be implemented in Intel Itanium-based computer systems and holds information from an MCA handler running from system firmware.

## Set Auxiliary Log Status

This command can be used by system software or firmware to set the status returned by the **Get Auxiliary Log Status** command. Some implementations might use a private mechanism to set this status, in which case this command can not be provided even if the **Get Auxiliary Log Status** command is provided.

## SDR Repository Device Commands

The following commands control the SDR repository device actions:

- [Get SDR Repository Info, page 7-5](#)
- [Get SDR Repository Allocation Info, page 7-6](#)
- [Reserve SDR Repository, page 7-6](#)
- [Get SDR, page 7-6](#)
- [Add SDR, page 7-6](#)
- [Partial Add SDR, page 7-6](#)
- [Delete SDR, page 7-6](#)
- [Clear SDR Repository, page 7-6](#)
- [Get SDR Repository Time, page 7-7](#)
- [Set SDR Repository Time, page 7-7](#)
- [Enter SDR Repository Update Mode, page 7-7](#)
- [Exit SDR Repository Update Mode, page 7-7](#)
- [Run Initialization Agent, page 7-7](#)

## Get SDR Repository Info

This command returns the SDR command version for the SDR repository. It also returns a timestamp for the last add, delete, or clear commands.

## Get SDR Repository Allocation Info

This command returns the number of possible allocation units, the amount of usable free space (in allocation units), the allocation unit size (in bytes), and the size of the largest contiguous free region (in allocation units). The allocation unit size is the number of bytes in which storage is allocated. For example, if a 20 byte record is to be added, and the SDR repository has a 16 byte allocation unit size, then the record would take up 32 bytes of storage.

## Reserve SDR Repository

This command sets the present owner of the repository, as identified by the software ID or the requester slave address from the command. The reservation process provides a limited amount of protection on repository access from the IPMB when records are being deleted or incrementally read.

## Get SDR

This command returns the sensor record specified by the record ID. The command also accepts a byte range specification that allows a selected portion of the record to be retrieved (incremental read). The **Reserve SDR Repository** command must be issued first for an incremental read to an offset other than 0000h. (The **Get SDR Repository Info** command should be used to verify the version of the SDR repository before sending other SDR repository commands. The command format and operation could change between versions.)

## Add SDR

This command adds the specified sensor record to the SDR repository and returns its record ID. The data passed in the request must contain all of the SDR data.

## Partial Add SDR

This command is a version of the **Add SDR** command that allows the record to be incrementally added to the repository. This command must be preceded by a **Reserve SDR Repository** command. The first partial add must be to offset 0000h, and partial adds must be done sequentially, with no gaps or overlap between the adds.

## Delete SDR

This command deletes the sensor record specified by record ID. The requester ID and the reservation ID must also match the owner of the SDR repository.

## Clear SDR Repository

This command clears all records from the SDR repository and reinitializes the SDR repository subsystem. The requestor ID and reservation ID information must match the present owner of the SDR repository. We recommend that this command not be used within your utilities and system management software.

## Get SDR Repository Time

This command returns the time setting from the SDR repository device, which the SDR repository devices uses for tracking when changes to the SDR repository are made.

## Set SDR Repository Time

This command initializes the time setting in the SDR repository device, which the SDR repository devices uses for tracking when changes to the SDR repository are made.

## Enter SDR Repository Update Mode

This command enters a mode that allows a subset of normal commands. Available commands are **Get Device ID**, **Get SDR**, **Add SDR**, **Partial Add SDR** and **Clear SDR Repository**.

## Exit SDR Repository Update Mode

This command exits the SDR repository update mode and restores normal use of all commands.

## Run Initialization Agent

This command runs the initialization agent and can also check the status of the agent.

## Event Receiver Commands

The following commands can be executed on the event receiver device:

- [Set Event Receiver, page 7-7](#)
- [Get Event Receiver, page 7-7](#)
- [Platform Event Message, page 7-7](#)

## Set Event Receiver

This is a global command to tell a controller where to send event messages. The slave address and LUN of the event receiver must be provided. A value FFh for the event receiver slave address disables the generation of event messages.

## Get Event Receiver

This is a global command to retrieve the present setting for the event receiver slave address and LUN.

## Platform Event Message

This command is a request for the BMC to process event data that the command contains. The data is logged to the SEL.

## SEL Record Examples

Examples that are reported to the SEL Repository are provided here. The raw record contains 16 bytes and are displayed in the examples as hexadecimal values. Following the arrow is the translation of the data. The l-pipes are separators for ease of reading the translation.

The following topics are included:

- [Device Presence Changes, page 7-8](#)
- [LED Color Changes, page 7-9](#)
- [Voltage Changes, page 7-9](#)
- [Temperature Changes, page 7-9](#)

## Device Presence Changes

These are examples of presence assertions. This shows a boot-up process.

```

54 01 02 3c 0c 00 00 01 00 04 12 83 6f 01 ff 00 -----> 154 | 01/01/1970 00:52:12
| BIOS | System Event #0x83 | OEM System Boot Event | | Asserted
55 01 02 3d 0c 00 00 20 00 04 25 53 08 01 ff ff -----> 155 | 01/01/1970 00:52:13
| BMC | Entity presence BIOS_POST_CMPLT #0x53 | Device Present | Asserted
56 01 02 54 0c 00 00 20 00 04 25 52 08 00 ff ff -----> 156 | 01/01/1970 00:52:36
| BMC | Entity presence MAIN_POWER #0x52 | Device Absent | Asserted
57 01 02 25 00 00 00 20 00 04 25 41 08 01 ff ff -----> 157 | 01/01/1970 00:00:37
| BMC | Entity presence MEZZ_PRS #0x41 | Device Present | Asserted
58 01 02 25 00 00 00 20 00 04 25 43 08 00 ff ff -----> 158 | 01/01/1970 00:00:37
| BMC | Entity presence HDD1_PRS #0x43 | Device Absent | Asserted
59 01 02 25 00 00 00 20 00 04 25 45 08 01 ff ff -----> 159 | 01/01/1970 00:00:37
| BMC | Entity presence P1_PRESENT #0x45 | Device Present | Asserted
5a 01 02 25 00 00 00 20 00 04 25 47 08 00 ff ff -----> 15a | 01/01/1970 00:00:37
| BMC | Entity presence DDR3_P2_D2_PRS #0x47 | Device Absent | Asserted
5b 01 02 25 00 00 00 20 00 04 25 49 08 00 ff ff -----> 15b | 01/01/1970 00:00:37
| BMC | Entity presence DDR3_P2_E2_PRS #0x49 | Device Absent | Asserted
5c 01 02 25 00 00 00 20 00 04 25 4b 08 00 ff ff -----> 15c | 01/01/1970 00:00:37
| BMC | Entity presence DDR3_P2_F2_PRS #0x4b | Device Absent | Asserted
5d 01 02 26 00 00 00 20 00 04 25 4d 08 00 ff ff -----> 15d | 01/01/1970 00:00:38
| BMC | Entity presence DDR3_P1_A2_PRS #0x4d | Device Absent | Asserted
5e 01 02 26 00 00 00 20 00 04 25 4f 08 00 ff ff -----> 15e | 01/01/1970 00:00:38
| BMC | Entity presence DDR3_P1_B2_PRS #0x4f | Device Absent | Asserted
5f 01 02 26 00 00 00 20 00 04 25 51 08 00 ff ff -----> 15f | 01/01/1970 00:00:38
| BMC | Entity presence DDR3_P1_C2_PRS #0x51 | Device Absent | Asserted
60 01 02 26 00 00 00 20 00 04 25 53 08 01 ff ff -----> 160 | 01/01/1970 00:00:38
| BMC | Entity presence BIOS_POST_CMPLT #0x53 | Device Present | Asserted

```



## LED Color Changes

These are examples of LED color changes written into the SEL Repository.

```

34 05 02 2f 00 00 00 20 00 04 24 56 7f 00 04 10 -----> 534 | 01/01/1970 00:00:47
| BMC | Platform alert LED_MEZZ_TP_FLT #0x56 | LED is off | Asserted
35 05 02 30 00 00 00 20 00 04 24 56 7f 07 04 10 -----> 535 | 01/01/1970 00:00:48
| BMC | Platform alert LED_MEZZ_TP_FLT #0x56 | LED color is red | Asserted
36 05 02 30 00 00 00 20 00 04 24 58 7f 00 04 10 -----> 536 | 01/01/1970 00:00:48
| BMC | Platform alert LED_SYS_ACT #0x58 | LED is off | Asserted
37 05 02 31 00 00 00 20 00 04 24 58 7f 04 04 10 -----> 537 | 01/01/1970 00:00:49
| BMC | Platform alert LED_SYS_ACT #0x58 | LED color is green | Asserted
38 05 02 31 00 00 00 20 00 04 24 5a 7f 00 04 10 -----> 538 | 01/01/1970 00:00:49
| BMC | Platform alert LED_SAS1_FAULT #0x5a | LED is off | Asserted
39 05 02 32 00 00 00 20 00 04 24 5a 7f 05 04 10 -----> 539 | 01/01/1970 00:00:50
| BMC | Platform alert LED_SAS1_FAULT #0x5a | LED color is amber | Asserted

```

## Voltage Changes

These are examples of SEL messages when voltage thresholds are crossed.

```

7b 09 02 3d 19 00 00 20 00 04 02 00 01 52 b5 b7 -----> 97b | 01/01/1970 01:47:41
| BMC | Voltage P3V_BAT_SCALED #0x00 | Lower critical - going low | Asserted | Reading
2.39 < Threshold 2.42 Volts
8d 09 02 5b 19 00 00 20 00 04 02 00 81 52 bc b7 -----> 98d | 01/01/1970 01:48:11
| BMC | Voltage P3V_BAT_SCALED #0x00 | Lower critical - going low | Deasserted | Reading
2.48 > Threshold 2.42 Volts

```

## Temperature Changes

These are examples of SEL messages when temperature thresholds are crossed.

```

00 02 02 2b 00 00 00 20 00 04 19 18 05 00 ff ff -----> 200 | 01/01/1970 00:00:43
| BMC | Chip Set IOH_THERMTRIP_N #0x18 | Limit Not Exceeded | Asserted
12 02 02 31 00 00 00 20 00 04 07 19 05 00 ff ff -----> 212 | 01/01/1970 00:00:49
| BMC | Processor P2_THERMTRIP_N #0x19 | Limit Not Exceeded | Asserted
13 02 02 32 00 00 00 20 00 04 07 1a 05 00 ff ff -----> 213 | 01/01/1970 00:00:50
| BMC | Processor P1_THERMTRIP_N #0x1a | Limit Not Exceeded | Asserted

```

