

**Intel® Optane™ DC Persistent  
Memory Software Specification  
for UEFI\*  
*Revision 1.17***

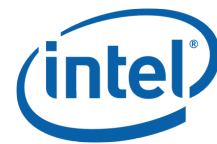
**Intel Confidential**

Reference Number 598513

# Table of Contents

Preface .....	1
Legal .....	1
<b><u>Revision History</u></b> .....	2
1.17 .....	2
1.16 .....	2
1.15 .....	2
1.14 .....	2
1.13 .....	3
1.12 .....	3
Supporting Documents .....	3
1. Introduction .....	4
1.1. Document Formatting .....	4
1.2. Launching the CLI .....	4
1.3. Command Syntax .....	4
1.4. DIMM Identification .....	5
1.5. Capacities .....	7
1.6. DCPMM Long Operations (DEVICE BUSY) .....	7
1.7. DCPMMs Must Be Present .....	7
1.8. Debug / Verbose .....	7
2. Commands .....	9
2.1. DIMM Discovery .....	9
2.1.1. Show Topology .....	9
2.1.2. Show Socket .....	12
2.1.3. Show Device .....	14
2.1.4. Show Memory Resources .....	27
2.1.5. Show System Capabilities .....	30
2.2. Provisioning .....	35
2.2.1. Create Memory Allocation Goal .....	36
2.2.2. Show Memory Allocation Goal .....	40
2.2.3. Dump Memory Allocation Settings .....	44
2.2.4. Load Memory Allocation Goal .....	46
2.2.5. Delete Memory Allocation Goal .....	49
2.3. Security .....	50
2.3.1. Enable Device Security .....	51

2.3.2. Change Device Passphrase .....	54
2.3.3. Change Device Security .....	56
2.3.4. Erase Device Data .....	58
2.4. Instrumentation .....	60
2.4.1. Show Sensor .....	60
2.4.2. Change Sensor Settings .....	64
2.5. Persistent Memory Provisioning .....	66
2.5.1. Show Persistent Memory .....	67
2.5.2. Show Namespace .....	70
2.5.3. Create Namespace .....	73
2.5.4. Modify Namespace .....	75
2.5.5. Delete Namespace .....	77
2.6. Support and Maintenance .....	79
2.6.1. Help .....	79
2.6.2. Version .....	80
2.6.3. Show Device Firmware .....	81
2.6.4. Update Firmware .....	83
2.6.5. Show Preferences .....	87
2.6.6. Change Preferences .....	89
2.7. Debug .....	91
2.7.1. Run Diagnostic .....	91
2.7.2. Show Error Log .....	93
2.7.3. Dump Debug Log .....	95
2.7.4. Show ACPI Tables .....	97
2.7.5. Show Device Platform Configuration Data .....	98
2.7.6. Delete Device Platform Configuration Data .....	99
2.7.7. Show SMBIOS Tables .....	102



# Preface

## Legal

Notice: This document contains information on products in the design phase of development. The information here is subject to change without notice. Do not finalize a design with this information.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software, or service activation. Learn more at [intel.com](http://intel.com), or from the OEM or retailer.

No computer system can be absolutely secure. Intel does not assume any liability for lost or stolen data or systems or any damages resulting from such losses.

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

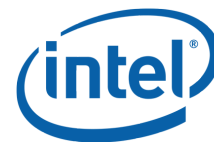
This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice.

Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting [www.intel.com/design/literature.htm](http://www.intel.com/design/literature.htm).

Intel, the Intel logo, Intel Optane, and Xeon are trademarks of Intel Corporation in the



U. S. and/or other countries.

\*Other names and brands may be claimed as the property of others.

Copyright © 2019. Intel Corporation. All rights Reserved.

## Revision History

### 1.17

- Added notes to delete-pcd, load-goal and create-goal commands
- Fixed some errors in text

### 1.16

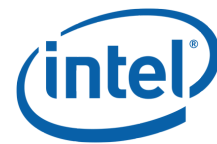
- Added clarifying note to State field of command [Run Diagnostic](#)

### 1.15

- Added clarifying text to Show Namespaces for HealthStates.
- Added clarifying text to Show Device for HealthStates.
- Corrected text in Create Goal to indicate Reserved capacity will be presented as Reserved in Show Memory Resources.
- Added note to clarify Create Goal capacity values are a target goal and may be different from actual applied by platform firmware.
- Split Memory Subsystem Provisioning into 2 sections: Provisioning and Security.
- Added note to clarify Dump Debug Log may collect logs via SMBus for non-functional DCPMM's.
- Create Namespace: A default volume name will not be generated if not explicitly requested.

### 1.14

- Added section ***DIMM Identification***
- Minor text clarifications
- Updated Show Memory Resources definitions



- Added note to Update Firmware indicating TSOD must be disabled for recover operations.

## 1.13

- Fixed missing section and misc formatting

## 1.12

- Added BootStatusRegister to Show Device
- Added asciidoc support
- Styling changes, added logo, modified theme, page/version headers
- Added ControllerRevisionID to show device
- Added new Playback/Record section
- Added clarification that long-operations prevent security state changes
- Changed documented behavior for firmware update: continue on error
- Updated all commands with dimm target so values are optional; treat as ALL if not specified
- Added explicit notes to sections to clearly state changes to persistent memory (goal/namespace/pcd) do not delete/modify user data
- FIS 1.13 changes
  - Dump Debug Log: Will attempt to retrieve FW logs from 3 sources
  - Show Device:
    - LastShutdownStatus now Latched/UnlatchedLastShutdownStatus
    - Added MaxAveragePowerBudget attribute

## Supporting Documents

- Intel® NVM Dimm Management Software-Firmware Interface Specification (554415)
- Intel® Optane™ DC Persistent Memory Module Firmware Interface Specification (556488)



# 1. Introduction

This document covers UEFI command line access utility. The Intel® Optane™ DC Persistent Memory Module (DCPMM) UEFI Command Line Interface (CLI) is a manufacturing and validation utility providing access to a subset of the management features of DCPMMs in the UEFI environment.

## 1.1. Document Formatting

Throughout this specification, the syntax of each command is documented in a shaded text box. Items in [brackets] are optional. For options, targets and properties, each possible value is separated by a '|' meaning "or" and the default value is italicized. Items in (parenthesis) indicate a user supplied value. For example, the following command syntax specifies that the verb "set" is required followed by an optional option "help". The target *-example* is required and a value for ExampleID can optionally be provided. It also specifies an optional property "Test" in which valid values are "Test1" or "Test2" with Test1 being the default.

```
ipmctl set [-h|-help] -example [(ExampleID)]  
[Test=(Test1|Test2)]
```

## 1.2. Launching the CLI

The CLI application is launched from the OS/UEFI shell. To run a single command, supply the command and parameters immediately following the binary name. The command will run, display the resulting data and return control to the shell. If no command is provided, the CLI will display the help information as documented in the Show Help command.

```
ipmctl (command)
```

## 1.3. Command Syntax

The command line syntax is case insensitive and is interpreted in English only. It follows the DMTF SM CLP standard (DSP0214) with the exception of the target portion of the command. Generally the form of a user request is:

```
ipmctl <verb>[<options>][<targets>][<properties>]
```



A command has a single verb which represents the action to be taken. Following the verb are zero or more options which modify the action of the verb; overriding the default behavior with explicitly requested behavior. Options generally have a short and long form (e.g., -a | -all). Generally one or more targets are required to indicate the object of the action. However, there are a few cases where a target is not required. Finally, zero or more properties defined as a key/value pair can be used to modify the target. The ordering of the command parts shown above is enforced (e.g., a target cannot precede an option). However, ordering within the command parts is not enforced (e.g., options can be provided in any order, targets can be provided in any order and properties can be provided in any order). For example, the following command will generate a syntax error because the option -all comes after the target -system and therefore the command parts are mis-ordered (verb then target then option).

```
ipmctl show -system -all
```

Whereas, these commands are both valid even though the options are specified in different order because the command parts are provided in the correct order (verb then options then target).

```
ipmctl show -help -all -system  
ipmctl show -all -help -system
```

If an option or target allows input of more than one value, it is expected as a comma separated list without whitespace between the values. For example, "-display value1,value2" instead of "-display value1, value2".

Properties follow the syntax [PROPERTIES]=Value.

## 1.4. DIMM Identification

Throughout this document, DCPMMs are referenced by one of two unique IDs: DimmHandle & DimmUID. Either ID may be used for commands that utilize a *-dimm* target.

For example, each of the following are equivalent:





```
ipmctl show -d DimmHandle,DimmUID -dimm 8089-a2-1748-00000001
ipmctl show -d DimmHandle,DimmUID -dimm 0x0001
ipmctl show -d DimmHandle,DimmUID -dimm 1
---DimmID=0x0001---
    DimmHandle=0x0001
    DimmUID=8089-a2-1748-00000001
```

The *-dimm* target also accepts a comma separated list.

```
ipmctl show -d DimmHandle,DimmUID -dimm
0x0001,0x1001
ipmctl show -d DimmHandle,DimmUID -dimm
0x0001,8089-a2-1748-00000002
---DimmID=0x0001---
    DimmHandle=0x0001
    DimmUID=8089-a2-1748-00000001
---DimmID=0x1001---
    DimmHandle=0x1001
    DimmUID=8089-a2-1748-00000002
```

The default DimmID output format can be modified by changing the `CLI_DEFAULT_DIMM_ID` property of using the [Change Preferences](#) command. For example, with `CLI_DEFAULT_DIMM_ID=UID`:

```
ipmctl show -d DimmHandle,DimmUID -dimm 0x0001
---DimmID=8089-a2-1748-00000001---
    DimmHandle=0x0001
    DimmUID=8089-a2-1748-00000001
```

## DimmHandle

The DCPMM handle is formatted as 0xABCD.

- A = Socket
- B = Memory Controller
- C = Channel
- D = Slot

## DimmUID

The unique identifier of the DCPMM formatted as VVVV-ML-MMYYSNSNSNSN or VVVV-SNSNSNSN (if the manufacturing information is not available) where:

- VVVV = VendorID



- ML = ManufacturingLocation
- MMY = ManufacturingDate
- SNSNSNSN = SerialNumber

## 1.5. Capacities

Capacities can be displayed in bytes (B), binary multiples of bytes (MiB, GiB, TiB), or decimal multiples of bytes (MB, GB, TB) where: \* 1 mebibyte (MiB) =  $1024^2$  bytes \* 1 gibibyte (GiB) =  $1024^3$  bytes \* 1 tebibyte (TiB) =  $1024^4$  bytes \* 1 megabyte (MB) =  $1000^2$  bytes \* 1 gigabyte (GB) =  $1000^3$  bytes \* 1 terabyte (TB) =  $1000^4$  bytes By default, capacities, except bytes, are displayed to 1/10 precision. Therefore, a capacity may be displayed as 0.0 if the value is less than 0.1. The default display of capacities is configurable with the command [Change Preferences](#). The input and display of capacities in an individual command can also be overridden by supplying the -units option. Capacities can be input in either decimal or hexadecimal format with a '0x' prefix.

## 1.6. DCPMM Long Operations (DEVICE BUSY)

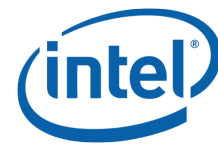
DCPMMs may be busy while processing some operations. These are defined as 'Long Operations' in the Firmware Interface Specification (FIS). While a long operation is in progress, some requests may be rejected as indicated by 'DEVICE BUSY' status. Commands may be retried and are expected to succeed when the long operation is complete. Show Device may be used to determine if ARS (ARSStatus) or Overwrite DIMM (OverwriteStatus) long operations are in progress.

## 1.7. DCPMMs Must Be Present

If DCPMMs are not present then the driver will not load and related functionality via the driver protocol and CLI will be unavailable.

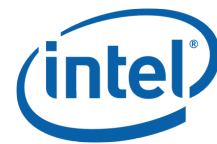
## 1.8. Debug / Verbose

The CLI provides the capability to view debug messages when executing a command by specifying the verbose mode output option. This is intended to aid diagnosing a problem with the CLI itself and may be useful when submitting a defect.



Changes the output format to XML and view debug message for the command.

```
ipmctl show -verbose -dimm
```



## 2. Commands

### 2.1. DIMM Discovery

**DIMM Identifiers** The default DCPMM identifier "DimmID" is equivalent to DimmHandle as described in the command [Show Device](#). This may be changed to DimmUID using the command [Change Preferences](#). In all commands accepting the -dimm target as input, either the DimmHandle or the DimmUID may be used to identify a DCPMM.

**DIMM Manageability** Beyond discovery, most DCPMM operations require the ability of host software to manage the DCPMM. Manageability is determined by the interface format code, the vendor identifier, device identifier and the firmware API version and can be retrieved with the command Section [Show Device](#).

**DIMM Compatibility** If the host software detects that the system is populated with DCPMMs with incompatible SKUs or the DCPMMs are configured in violation of their license, it will indicate such and operate in read-only mode. In this case, the host software will not allow changes to the DCPMMs and their associated capacity.

**Limitations** If there are no DCPMMs present, then associated UEFI driver is not loaded, and none of the commands (except version) are available.

#### 2.1.1. Show Topology

##### NAME

ipmctl-show-topology - Shows the topology of the memory installed

##### SYNOPSIS

```
ipmctl show [OPTIONS] -topology [TARGETS]
```

##### DESCRIPTION

Shows the topology of the memory installed in the host server. Use the command ipmctl-show-device to view more detailed information about a DCPMM.

##### OPTIONS



**-a**

**-all**

Shows all attributes.

**NOTE**

The all and display options are exclusive and may not be used together.

**-d (attributes)**

**-display (attributes)**

Filters the returned attributes by explicitly specifying a comma-separated list of any of the attributes defined in the Return Data section.

**NOTE**

The all and display options are exclusive and may not be used together.

**-h**

**-help**

Displays help for the command.

**-u (B | MB | MiB | GB | GiB | TB | TiB)**

**-units (B | MB | MiB | GB | GiB | TB | TiB)**

Changes the units that capacities are displayed in for this command. One of: bytes (B), megabytes (MB), mebibytes (MiB), gigabytes (GB), gibibytes (GiB), terabytes (TB) or tebibytes (TiB).

## TARGETS

**-dimm [(DimmIDs)]**

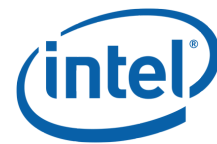
Restricts output to specific DIMMs by optionally supplying the DIMM target and one or more comma-separated DIMM identifiers. The default is to display all DIMMs.

**-socket (SocketIDs)**

Restricts output to the DIMMs installed on specific sockets by supplying the socket target and one or more comma-separated socket identifiers. The default is to display all sockets.

**NOTE**

If ACPI PMTT table is not present, then DDR4 memory will not be displayed in the filtered socket list.



## EXAMPLES

Displays the system memory topology.

```
ipmctl show -topology
```

## RETURN DATA

Displays a table with the attributes listed below for each memory DIMM installed in the host server.

### MemoryType

(Default) The DIMM type. One of:

- Unknown
- DDR4
- Logical Non-volatile Device

### Capacity

(Default) The raw capacity of the DIMM as reported in the SMBIOS Type 17 table.

### DimmID

(Default) The DIMM identifier. For DRAM DIMMs, the DimmID is "N/A".

### PhysicalID

(Default) The DIMM physical identifier (i.e., SMBIOS Type 17 handle).

### DeviceLocator

(Default) The string that identifies the physically-labeled socket or board position where the DIMM is located.

### SocketID

The processor socket identifier (i.e., NUMA node) where the DIMM is installed. For DRAM DIMMs, the socket identifier is "N/A".

### MemControllerID

The associated memory controller identifier. For DRAM DIMMs, the memory controller identifier is "N/A".

### ChannelID

The associated channel. For DRAM DIMMs, the channel identifier is "N/A".

### ChannelPos



The DIMM position in the channel. For DRAM DIMMs, the channel position identifier is "N/A".

#### **NodeControllerID**

The node controller identifier. For DRAM DIMMs, the node controller identifier is "N/A".

#### **BankLabel**

The string that identifies the physically-labeled bank where the DIMM is located.

## 2.1.2. Show Socket

### NAME

ipmctl-show-socket - Shows basic information about the physical processors

### SYNOPSIS

```
ipmctl show [OPTIONS] -socket [TARGETS]
```

### DESCRIPTION

Shows basic information about the physical processors in the host server.

### OPTIONS

**-a**

**-all**

Shows all attributes.

#### NOTE

The all and display options are exclusive and may not be used together.

**-d (attributes)**

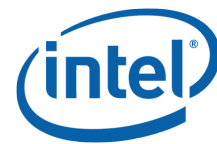
**-display (attributes)**

Filters the returned attributes by explicitly specifying a comma-separated list of any of the attributes defined in the Return Data section.

**-h**

**-help**

Displays help for the command.



**-u (B | MB | MiB | GB | GiB | TB | TiB)**

**-units (B | MB | MiB | GB | GiB | TB | TiB)**

Changes the units that capacities are displayed in for this command. One of: bytes (B), megabytes (MB), mebibytes (MiB), gigabytes (GB), gibibytes (GiB), terabytes (TB) or tebibytes (TiB).

## TARGETS

**-socket (SocketIDs)**

Restricts output to the DIMMs installed on specific sockets by supplying the socket target and one or more comma-separated socket identifiers. The default is to display all sockets.

## EXAMPLES

Displays information about all the processors.

```
ipmctl show -socket
```

Lists all properties for socket 1.

```
ipmctl show -socket 1
```

Retrieves specific properties for each processor.

```
ipmctl show -d MappedMemoryLimit -socket
```

## RETURN DATA

Displays a table with the attributes listed below for each physical processor installed in the host server.

### SocketID

(Default) The processor socket identifier.

### MappedMemoryLimit

(Default) The maximum amount of memory that is allowed to be mapped into the system physical address space for this processor based on its SKU.

### TotalMappedMemory

(Default) The total amount of memory that is currently mapped into the system physical address space for this processor.





## 2.1.3. Show Device

### NAME

ipmctl-show-device - Shows information about one or more DCPMMs

### SYNOPSIS

```
ipmctl show [OPTIONS] -dimm [TARGETS]
```

### DESCRIPTION

Shows information about one or more DCPMMs.

### OPTIONS

**-a**

**-all**

Shows all attributes.

#### NOTE

The all and display options are exclusive and may not be used together.

**-d (attributes)**

**-display (attributes)**

Filters the returned attributes by explicitly specifying a comma-separated list of any of the attributes defined in the Return Data section.

#### NOTE

The all and display options are exclusive and may not be used together.

**-h**

**-help**

Displays help for the command.

**-u (B|MB|MiB|GB|GiB|TB|TiB)**

**-units (B|MB|MiB|GB|GiB|TB|TiB)**

Changes the units that capacities are displayed in for this command. One of: bytes (B), megabytes (MB), mebibytes (MiB), gigabytes (GB), gibibytes (GiB), terabytes (TB) or tebibytes (TiB).



## TARGETS

### **-dimm (DimmIDs)**

Restricts output to specific DCPMMs by supplying the DIMM target and one or more comma-separated DCPMM identifiers. The default is to display all DCPMMs.

### **-socket (SocketIDs)**

Restricts output to the DCPMMs installed on specific sockets by supplying the socket target and one or more comma-separated socket identifiers. The default is to display all sockets.

#### **NOTE**

If ACPI PMTT table is not present, then DDR4 memory will not be displayed in the filtered socket list.

## EXAMPLES

Lists a few key fields for each DCPMM.

```
ipmctl show -dimm
```

Lists all properties for DCPMM 0x0001.

```
ipmctl show -a -dimm 0x0001
```

Retrieves specific properties for each DCPMM.

```
ipmctl show -d HealthState,LockState -dimm
```

## RETURN DATA

The default behavior is to display a table with the default attributes listed below; applying options changes the output to a more detailed format. Limited information (noted in the table below) is applicable if the DCPMM is not manageable by the software as indicated by the "ManageabilityState" property.

#### **NOTE**

Some data is endian-swapped for human readability.

### **DimmID**

(Default) The DCPMM identifier.

### **Capacity**

(Default) The usable capacity of the DCPMM as reported by the firmware.



## HealthState

(Default) Overall DCPMM health. One of:

- Healthy
- Noncritical: Maintenance is required.
- Critical: Features or performance are degraded due to failure.
- Fatal: Data loss has occurred or is imminent. In this case, the firmware will disable the media and access to user data and operations that require use of the media will fail.
- Non-functional: The DCPMM is present but is non-responsive via the DDRT communication path. It may be possible to communicate with this DCPMM via SMBus for a subset of commands.
- Unmanageable: The DCPMM is not supported by this version of the software
- Unknown: Unable to determine the DCPMM health state.

## HealthStateReason

Indicates why the DCPMM is in the current Health State. One or more of:

- None
- Percentage Remaining less than 1%
- Package Sparing occurred
- CAP Self-Test warning
- Percentage Remaining is 0
- Die Failure
- AIT DRAM disabled
- CAP Self-Test fail
- Critical internal failure

Refer to the *Intel® Optane™ DC Persistent Memory Firmware Interface Specification*, document number 556488, Section 4.8 SMART and Health.

## InterfaceFormatCode

A comma-delimited list of the JEDEC standard format interface code(s) for the DCPMM where each code is formatted as: code (JEDEC Description or "Unknown").

## ManageabilityState

Ability of the DCPMM host software to manage the DCPMM. Manageability is determined by the interface format code, the vendor identifier, device identifier and the firmware API version. One of:



- Manageable: The DCPMM is manageable by the software.
- Unmanageable: The DCPMM is not supported by this version of the software.

**PhysicalID**

The DCPMM physical identifier (i.e., SMBIOS Type 17 handle).

**DimmHandle**

The DCPMM handle formatted as 0xABCD.

- A = Socket
- B = Memory Controller
- C = Channel
- D = Slot

**DimmUID**

The unique identifier of the DCPMM formatted as VVVV-ML-MMYYSNSNSNSN or VVVV-SNSNSNSN (if the manufacturing information is not available) where:

- VVVV = VendorID
- ML = ManufacturingLocation
- MMY = ManufacturingDate
- SNSNSNSN = SerialNumber

**SocketID**

The processor socket identifier (i.e., NUMA node) where the DCPMM is installed.

**MemControllerID**

The associated memory controller identifier.

**ChannelID**

The associated channel.

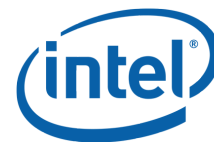
**ChannelPos**

The DCPMM position in the channel.

**MemoryType**

The memory type. One of:

- Unknown
- DDR4



- Logical Non-Volatile Device

**VendorID**

The vendor identifier of the DCPMM. This value is presented in big endian format.

**DeviceID**

The device identifier of the DCPMM. This value is presented in big endian format.

**RevisionID**

The revision identifier of the DCPMM.

**SubsystemVendorID**

The vendor identifier of the non-volatile memory subsystem controller. This value is presented in big endian format.

**SubsystemDeviceID**

The device identifier of the non-volatile memory subsystem controller.

**SubsystemRevisionID**

The revision identifier of the non-volatile memory subsystem controller retrieved from NFIT. This field uses a different encoding than ControllerRevisionID.

**ManufacturingInfoValid**

If the manufacturing location and date are valid. One of:

- 0: Not valid
- 1: Valid

**ManufacturingLocation**

The manufacturing location assigned by the vendor or "N/A" if ManufacturingInfoValid is 0.

**ManufacturingDate**

The manufacturing date assigned by the vendor or "N/A" if ManufacturingInfoValid is 0.

**SerialNumber**

The serial number assigned by the vendor. This value is presented in big endian format.

**PartNumber**

The part number assigned by the vendor

**DeviceLocator**

A string describing the physically-labeled socket or board position where the memory device is located from the SMBIOS Type 17 Memory Device table.

**BankLabel**

A string that identifies the physically labeled bank where the memory device is located from the SMBIOS Type 17 Memory Device table.

**DataWidth**

The width in bits used to store user data from the SMBIOS Type 17 Memory Device table.

**TotalWidth**

The width in bits for data and error correction and/or data redundancy from the SMBIOS Type 17 Memory Device table.

**Speed**

The maximum capable speed of the device in megahertz (MHz) from the SMBIOS Type 17 Memory Device table.

**FormFactor**

The DCPMM form factor (i.e., SMBIOS Type 17 Memory Device Form Factor). One of:

- Unknown
- DIMM
- SODIMM

**LockState**

(Default) The current security state of the persistent memory on the DCPMM. One of:

- Unknown - The security state cannot be determined (e.g., when the DCPMM is not manageable by the software).
- Disabled - Security is not enabled.
- Disabled, Frozen - Security is not enabled. A reboot is required to change the security state.
- Unlocked - Security is enabled and unlocked.
- Unlocked, Frozen - Security is enabled and unlocked. A reboot is required to change the security state.



- Locked - Security is enabled and locked.
- Exceeded - The passphrase limit has been reached. A power cycle is required to change the security state.
- Not Supported - Security is not supported on the DCPMM.

### **FWVersion**

(Default) The BCD-formatted revision of the active firmware in the format PN.RN.SV.bbbb where:

- PN = 2-digit product number
- RN = 2-digit revision number
- SN = 2-digit security revision number
- bbbb = 4-digit build version

Value may be N/A if the DCPMM is not manageable by the software.

### **FWAPIVersion**

The firmware supported interface revision in the format aa.bb where:

- aa = 2-digit major version
- bb = 2-digit minor version

The firmware interface is intended to be backwards compatible. Therefore, the host software allows management of DCPMMs where this version is less than or equal to the version stored in the host software. Value may be N/A if the DCPMM is not manageable by the software.

**The following information is only applicable when the DCPMM is manageable by the software as indicated by the "ManageabilityState".**

### **ManufacturerID**

The manufacturer identifier of the DCPMM. This value is presented in big endian format.

### **ControllerRevisionID**

The controller stepping and revision ID retrieved from the controller FW. This field uses a different encoding than SubsystemRevisionID.

### **IsNew**

Whether or not the DCPMM is incorporated with the rest of the DCPMM in the system. One of:



- 0: Configured
- 1: The DCPMM requires configuration.

**MemoryCapacity**

Usable DCPMM Memory Mode capacity.

**AppDirectCapacity**

Usable DCPMM App Direct capacity.

**UnconfiguredCapacity**

DCPMM capacity that is inaccessible because it is not mapped into the system physical address space.

**InaccessibleCapacity**

DCPMM capacity that is inaccessible due to:

- licensing issue
- platform configuration prevents accessing this capacity. For example, MemoryCapacity is configured and available on a DCPMM but MemoryMode is not enabled by BIOS.

**ReservedCapacity**

DCPMM capacity reserved for proper alignment.

**PeakPowerBudget**

If the DCPMM firmware power management policy is enabled, the power budget in mW used for instantaneous power. Refer to FIS for allowable range and default value.

**AvgPowerBudget**

If the DCPMM firmware power management policy is enabled, the power budget in mW used for average power. Refer to FIS for allowable range and default value.

**MaxAveragePowerBudget**

Maximum average power budget supported by the DCPMM.

**PackageSparingCapable**

Whether or not the DCPMM supports package sparing. One of:

- 0: False
- 1: True





### **PackageSparingEnabled**

Whether or not the DCPMM package sparing policy is enabled. One of:

- 0: Disabled
- 1: Enabled

### **PackageSparesAvailable**

The number of spare devices available for package sparing.

### **LatchedLastShutdownStatus**

The status of the last shutdown of the DCPMM. One or more of:

- Unknown: The last shutdown status cannot be determined.
- FW Flush Complete: FW flush completed.
- PM ADR Command: Power management ADR command received.
- PM S3: Power management S3 command received.
- PM S5: Power management S5 command received.
- DDRT Power Fail Command: DDR power fail command received.
- PMIC 12V/DDRT 1.2V Power Loss (PLI)
- PM Warm Reset: Power management warm reset received.
- Thermal Shutdown: Thermal shutdown triggered.
- Controller's FW State Flush Complete: Flush Completed.
- Viral Interrupt: Viral interrupt received.
- Surprise Clock Stop: Surprise clock stop received.
- Write Data Flush Complete: Write data flush completed.
- PM S4: Power management S4 command received.
- PM Idle: Power management idle received
- DDRT Surprise Reset: Surprise reset received

### **UnlatchedLastShutdownStatus**

The status of the last shutdown status of the DCPMM. Same fields as the Latched Last Shutdown Status with the only difference that LLS details on a dirty shutdown is logged, even if the Latch System Shutdown Status was not enabled. One or more of:

- Unknown: The last shutdown status cannot be determined.
- FW Flush Complete: FW flush completed.



- PM ADR Command: Power management ADR command received.
- PM S3: Power management S3 command received.
- PM S5: Power management S5 command received.
- DDRT Power Fail Command: DDR power fail command received.
- PMIC 12V/DDRT 1.2V Power Loss (PLI)
- PM Warm Reset: Power management warm reset received.
- Thermal Shutdown: Thermal shutdown triggered.
- Controller's FW State Flush Complete: Flush Completed.
- Viral Interrupt: Viral interrupt received.
- Surprise Clock Stop: Surprise clock stop received.
- Write Data Flush Complete: Write data flush completed.
- PM S4: Power management S4 command received.
- PM Idle: Power management idle received
- DDRT Surprise Reset: Surprise reset received

### **LastShutdownTime**

The time the system was last shutdown.

### **ModesSupported**

A list of the modes supported by the DCPMM. Refer to the command Show System Capabilities to determine the modes supported by the platform. One or more of:

- Memory Mode: DCPMMs act as system memory under the control of the operating system. In Memory Mode, any DDR in the platform will act as a cache working in conjunction with the DCPMM.
- App Direct: DCPMMs and DDR act as independent memory resources under direct load/store control of the application.

### **SecurityCapabilities**

The security features supported by the DCPMM. Zero or more of:

- Encryption: The DCPMM supports persistent memory encryption by setting a passphrase.
- Erase: The DCPMM is erasable.

### **MasterPassphraseEnabled**

This property indicates if master passphrase is enabled. If it is disabled, then it



cannot be enabled. One of:

- 0: Disabled - Cannot be enabled
- 1: Enabled - Master passphrase can be changed. Cannot be disabled.

### **ConfigurationStatus**

The status of the DCPMM memory configuration. One of:

- Valid: The configuration is valid.
- Not Configured: The DCPMM has not been configured.
- Failed - Bad configuration: The configuration is corrupt.
- Failed - Broken interleave: This DCPMM is part of an interleave set that is not complete.
- Failed - Reverted: The configuration failed and was reverted to the last known good configuration.
- Failed - Unsupported: The configuration is not compatible with the installed BIOS.
- Unknown: The configuration cannot be determined.

### **SKUViolation**

The configuration of the DCPMM is unsupported due to a license issue. One of:

- 0: False
- 1: True

### **ARSStatus**

The address range scrub (ARS) operation status for the DCPMM. The status is a reflection of the last requested ARS, but not necessarily within the current platform power cycle. One of:

- Unknown - The ARS operation status cannot be determined.
- Not started - An ARS operation has not started.
- In progress - An ARS operation is currently in progress.
- Completed - The last ARS operation has completed.
- Aborted - The last ARS operation was aborted.

### **OverwriteStatus**

The overwrite DIMM operation status for the DCPMM. One of:



- Unknown - The overwrite DIMM operation status cannot be determined.
- Not started - An overwrite DIMM operation was not started on the last boot.
- In progress - An overwrite DIMM operation is currently in progress.
- Completed - An overwrite DIMM operation completed and a reboot is required to use the DCPMM.

### **ViralPolicy**

Whether viral policies are enabled on the DCPMM. One of:

- 0: Disabled - This is the default.
- 1: Enabled - The persistent memory on the DCPMM will be put into read-only mode if the host operating system software detects an uncorrectable error situation and indicates a viral state in order to prevent the spread of damage.

### **ViralState**

Whether the DCPMM is currently viral. One of:

- 0: Not Viral
- 1: Viral - The viral policies of the DCPMM have switched the persistent memory to read-only mode due to the host operating system software detecting an uncorrectable error situation and indicating a viral state.

### **AitDramEnabled**

If the DCPMM AIT DRAM is enabled. One of:

- 0: Disabled - The device will suffer performance degradation if the AIT DRAM becomes disabled.
- 1: Enabled

### **BootStatus**

The initialization status of the DCPMM as reported by the firmware in the boot status register. One or more of:

- Unknown - The boot status register cannot be read.
- Success - No errors were reported during initialization.  
**The following statuses indicate that the media is not functional and therefore access to user data and operations that require use of the media will fail.**
- Media Not Ready - The firmware did not complete media training.
- Media Error - The firmware detected an error during media training.



- Media Disabled - The firmware disabled the media due to a critical issue.  
**The following statuses indicate that communication with the firmware is not functional.**
- FW Assert - The firmware reported an assert during initialization.

### **BootStatusRegister**

The raw hex value of the DCPMM Boot Status Register of the DCPMM

### **ErrorInjectionEnabled**

Error injection status.

- 0: Disabled - This is the default.
- 1: Enabled

### **MediaTemperatureInjectionEnabled**

Media temperature injection status.

- 0: Disabled - This is the default.
- 1: Enabled

### **SoftwareTriggersEnabled**

Software trigger status.

- 0: Disabled - This is the default.
- 1: At least one software trigger enabled.

### **SoftwareTriggersEnabledDetail**

Comma separated list of software triggers currently enabled. One or more of:

- None
- Package Sparing
- Fatal Error
- Percentage Remaining
- Dirty Shutdown

### **PoisonErrorInjectionsCounter**

This counter is incremented each time the set poison error is successfully executed.

### **PoisonErrorClearCounter**

This counter is incremented each time the clear poison error is successfully



executed.

#### **MediaTemperatureInjectionsCounter**

This counter is incremented each time the media temperature is injected.

#### **SoftwareTriggersCounter**

This counter is incremented each time a software trigger is enabled.

## **2.1.4. Show Memory Resources**

### **NAME**

`ipmctl-show-memory-resources` - Shows the total DCPMM memory resource allocation

### **SYNOPSIS**

```
ipmctl show [OPTIONS] -memoryresources
```

### **DESCRIPTION**

Shows the total DCPMM memory resource allocation across the host server.

### **OPTIONS**

**-h**

**-help**

Displays help for the command.

**-u (B | MB | MiB | GB | GiB | TB | TiB)**

**-units (B | MB | MiB | GB | GiB | TB | TiB)**

Changes the units that capacities are displayed in for this command. One of: bytes (B), megabytes (MB), mebibytes (MiB), gigabytes (GB), gibibytes (GiB), terabytes (TB) or tebibytes (TiB).

### **EXAMPLES**

Shows the DCPMM memory resource allocation.

```
ipmctl show -memoryresources
```



## RETURN DATA

Returns the default attributes listed below.

<b>NOTE</b>	Capacities from unmanageable DCPMMs are not included in the following aggregated totals.
-------------	--

### **Capacity**

Total system DCPMM capacity.

### **MemoryCapacity**

Total usable system DCPMM Memory Mode capacity.

### **AppDirectCapacity**

Total usable system DCPMM App Direct capacity.

### **UnconfiguredCapacity**

Total system DCPMM capacity that is unusable because it has not been configured.

### **ReservedCapacity**

Total system DCPMM persistent memory capacity that is reserved. This capacity is the persistent memory partition capacity (rounded down for alignment) less any App Direct capacity. Reserved capacity typically results from a Memory Allocation Goal request that specified the Reserved property. This capacity is not mapped to system physical address (SPA) space.

### **InaccessibleCapacity**

Total system DCPMM capacity that is inaccessible due any of:

- Platform configuration prevents accessing this capacity. e.g. MemoryCapacity is configured but MemoryMode is not enabled by platform FW (current Memory Mode is 1LM).
- Capacity is inaccessible because it is not mapped into the system physical address space (SPA). This is usually due to platform firmware memory alignment requirements.
- DCPMM configured capacity but SKU prevents usage. e.g. AppDirectCapacity but DCPMM SKU is MemoryMode only.

## DETAILS

DCPMMs are partitioned into Memory and Persistent partitions. ipmctl will align the Memory partition on a 1 GiB boundary with the Persistent partition consuming the remaining capacity. An exception: if DCPMM is configured for 100% Memory Mode



then Memory partition will consume 100% of the capacity, while Persistent partition will be 0 length. Any capacity that falls outside the Memory and Persistent partitions is InaccessibleCapacity and is not useable.

Platform firmware alignment restrictions may result in some capacity from the Memory and Persistent partitions not mapped to system physical address (SPA) space. This memory is considered InaccessibleCapacity and is not usable.

Definitions:

**Total Capacity (TC)**

Raw Capacity (total usable) reported by DCPMM DIMM Partition Info

**Memory Partition Capacity (MPC)**

Volatile Capacity reported by DCPMM DIMM Partition Info

**Persistent Partition Capacity (PPC)**

Persistent Capacity reported by DCPMM DIMM Partition Info

**Volatile Memory Size (VMS)**

Usable volatile memory capacity as reported by platform FW via *Intel NVDIMM Current Config* → *Volatile Memory Size Mapped into SPA* field

**Persistent Memory Size (PMS)**

Usable persistent memory capacity as reported by platform FW via *Intel NVDIMM Current Config* → *Persistent Memory Size Mapped into SPA* field

**DCPMM DIMM Partition Info**

DIMM Partition Info provided by DCPMM firmware. See *Intel® Optane™ DC Persistent Memory Module Firmware Interface Specification* for details.

**Intel NVDIMM Current Config**

See *Intel® NVM Dimm Management Software-Firmware Interface Specification* for details.

**Calculations:**

MemoryCapacity = Volatile Memory Size

AppDirectCapacity = Persistent Memory Size

ReservedCapacity = PPC (rounded down for PM alignment) - PMS





```
InaccessibleCapacity =  
    + (TC - MPC - PPC)  
    + (TC - PMS - ReservedCapacity)  
if (CurrentMode == 1LM) then  
    + VMS (rounded down for alignment)
```

## 2.1.5. Show System Capabilities

### NAME

ipmctl-show-system-capabilities - Shows the platform supported DCPMM capabilities

### SYNOPSIS

```
ipmctl show [OPTIONS] -system -capabilities
```

### DESCRIPTION

Shows the total DCPMM memory resource allocation across the host server.

### OPTIONS

**-a**

**-all**

Shows all attributes.

#### NOTE

The all and display options are exclusive and may not be used together.

**-d (attributes)**

**-display (attributes)**

Filters the returned attributes by explicitly specifying a comma-separated list of any of the attributes defined in the Return Data section.

#### NOTE

The all and display options are exclusive and may not be used together.

**-h**

**-help**

Displays help for the command.



**-u (B | MB | MiB | GB | GiB | TB | TiB)**

**-units (B | MB | MiB | GB | GiB | TB | TiB)**

Changes the units that capacities are displayed in for this command. One of: bytes (B), megabytes (MB), mebibytes (MiB), gigabytes (GB), gibibytes (GiB), terabytes (TB) or tebibytes (TiB).

## EXAMPLES

Displays the supported DCPMM capabilities

```
ipmctl show -system -capabilities
```

## RETURN DATA

The default behavior is to return the default attributes listed below; the options can be used to expand or restrict the output.

### PlatformConfigSupported

(Default) Whether the platform level configuration of DCPMMs can be modified with the host software. One of:

- 0: Changes must be made in the BIOS.
- 1: The command **Create Memory Allocation Goal** is supported. Alignment (Default) Capacity alignment requirement for all memory types as reported by the BIOS.

### AllowedVolatileMode

(Default) The volatile mode allowed as determined by BIOS setup. One of:

- 1LM: One-level volatile mode. All memory resources in the platform are independently accessible, and not captive of the other resources.
- Memory Mode: DCPMMs act as system memory under the control of the operating system. In Memory Mode, any DDR in the platform will act as a cache working in conjunction with the DCPMMs.
- Unknown: The allowed volatile mode cannot be determined.

### CurrentVolatileMode

(Default) The current volatile mode. One of:

- 1LM: One-level volatile mode. All memory resources in the platform are independently accessible, and not captive of the other resources.
- Memory Mode: DCPMMs act as system memory under the control of the



operating system. In Memory Mode, any DDR in the platform will act as a cache working in conjunction with the DCPMMs.

- Unknown: The current volatile mode cannot be determined.

### **AllowedAppDirectMode**

(Default) The App Direct mode allowed as determined by BIOS setup. One of:

- Disabled: App Direct support is currently disabled by the BIOS.
- App Direct: App Direct support is currently enabled by the BIOS.
- Unknown: The current App Direct support cannot be determined.

### **ModesSupported**

A list of DCPMM modes supported by the BIOS. Refer to the command **Show Device** to determine the modes supported by the individual DCPMMs. One or more of:

- 1LM: One-level volatile mode. All memory resources in the platform are independently accessible, and not captive of the other resources.
- Memory Mode: DCPMMs act as system memory under the control of the operating system. In Memory Mode, any DDR in the platform will act as a cache working in conjunction with the DCPMMs.
- App Direct: DCPMMs and DDR act as independent memory resources under direct load/store control of the application.

### **SupportedAppDirectSettings**

The BIOS supported list of App Direct interleave settings in the format:

- x[Way] - [(IMCSize) iMC x (ChannelSize) Channel] followed by the input format for the command **Change Preferences**: (ByOne | (IMCSize)\_(ChannelSize)).

### **RecommendedAppDirectSettings**

The BIOS recommended list of App Direct interleave settings in the format:

- x[Way] - [(IMCSize) iMC x (ChannelSize) Channel] followed by the input format for the command **Change Preferences**: (ByOne | (IMCSize)\_(ChannelSize)).

### **MinNamespaceSize**

The minimum allowed namespace size as reported by the driver.

### **AppDirectMirrorSupported**

If the BIOS supports App Direct mirroring. One of:



- 0: Not supported.
- 1: Supported

### **DimmSpareSupported**

If the BIOS supports DCPMM sparing. One of:

- 0: Not supported
- 1: Supported

### **AppDirectMigrationSupported**

If the BIOS supports App Direct migration. One of:

- 0: Not supported
- 1: Supported

### **RenameNamespaceSupported**

If the host software supports renaming a namespace with the command Section [Modify Namespace](#). One of:

- 0: Not supported
- 1: Supported

### **GrowAppDirectNamespaceSupported**

If the host software supports increasing the capacity of an App Direct namespace with the command Section [Modify Namespace](#). One of:

- 0: Not supported
- 1: Supported

### **ShrinkAppDirectNamespaceSupported**

If the host software supports decreasing the capacity of an App Direct namespace with the command Section [Modify Namespace](#). One of:

- 0: Not supported
- 1: Supported

### **InitiateScrubSupported**

If the platform and host software support initiating an address range scrub on the DCPMMs in the system. One of:

- 0: Not supported



- 1: Supported

### **AdrSupported**

Whether the platform supports asynchronous DRAM refresh (ADR). One of:

- 0: Not supported. If ADR is not supported, App Direct data integrity cannot be assured during system interruptions.
- 1: Supported

### **EraseDeviceDataSupported**

Whether Erase Device Data is supported.

- 0: Not supported
- 1: Supported

### **EnableDeviceSecuritySupported**

Whether Enable Device Security is supported.

- 0: Not supported
- 1: Supported

### **DisableDeviceSecuritySupported**

Whether Change Device Security property Lockstate = Disabled is supported.

- 0: Not supported
- 1: Supported

### **UnlockDeviceSecuritySupported**

Whether Change Device Security property Lockstate = Unlocked is supported.

- 0: Not supported
- 1: Supported

### **FreezeDeviceSecuritySupported**

Whether Change Device Security property Lockstate = Frozen is supported.

- 0: Not supported
- 1: Supported

### **ChangeDevicePassphraseSupported**

Whether Change Device Passphrase is supported.



- 0: Not supported
- 1: Supported

#### **ChangeMasterPassphraseSupported**

Whether Change Master Passphrase is supported.

- 0: Not supported
- 1: Supported

#### **MasterEraseDeviceDataSupported**

Whether Master Erase Device Data is supported.

- 0: Not supported
- 1: Supported

## 2.2. Provisioning

Next generation memory controller and media technologies provide a number of opportunities to optimize a server's memory subsystem for a given workload. Three such opportunities are supported by the CLI and are covered in this section: memory tiering, reliability and performance options, and data-at-rest security options.

A system with both DRAM and DCPMMs can be configured to use memory tiering. The CLI supports configuring a "near" performance tier and a "far" capacity tier. In this two level memory system the near tier acts as a cache for the far tier. When memory tiering is enabled, DRAM DIMMs are utilized as near memory. The near memory tier utilizes all available DRAM capacity. The far memory tier is created from DCPMMs. Because the capacity of a DCPMM is so large (relative to DRAM) it may not be necessary to use all available DCPMM capacity as far memory (a typical near to far ratio is 1:8). So in addition to enabling memory tiering, the CLI provides for designating the amount of DCPMM capacity to be used as far memory. The remaining capacity can be used as App Direct persistent memory. Changing the tiering configuration on a running system is generally not supported. The CLI supports the creation of a tiering "goal configuration" which will take effect after a system reboot. A valid goal migrates into Memory Mode capacity and one or more persistent memory regions after it is successfully parsed and activated by BIOS on the next boot. See the command [Create Memory Allocation Goal](#) for more details on configuring the system for memory tiering.

Since modifying the memory tiering and persistent memory configuration requires a reboot to take affect it is possible to have both a current configuration and a pending "goal" configuration. Once the system is rebooted the goal configuration is applied



and becomes the current configuration. Prior to rebooting it's possible to see the goal configuration using the command Show Memory Allocation Goal. It is also possible to delete a goal prior to the reboot. See the command Delete Memory Allocation Goal for more details.

<b>WARNING</b>	<b>This command may result in data loss. Data should be backed up to other storage before executing this command.</b>
----------------	---

<b>NOTE</b>	Changing a memory allocation goal modifies how the platform firmware maps persistent memory in the system address space (SPA) which may result in data loss or inaccessible data, but does not explicitly delete or modify user data found in persistent memory.
-------------	--

## 2.2.1. Create Memory Allocation Goal

### NAME

ipmctl-create-goal - Creates a memory allocation goal on one or more DCPMM

### SYNOPSIS

```
ipmctl create [OPTIONS] -goal [TARGETS]
[PROPERTIES]
```

### DESCRIPTION

Creates a memory allocation goal on one or more for the BIOS to read on the next reboot in order to map the DCPMM capacity into the system address space. Persistent memory can then be utilized by creating a namespace.

<b>NOTE</b>	The capacity values presented by this command are a target goal or request to platform firmware. The actual capacity values are subject to change due to rounding and alignment requirements. If the goal request is invalid or not possible it may be rejected by platform firmware.
-------------	---

<b>NOTE</b>	Deleting the PCD can be used as a way to prepare individual DCPMMs for provisioning. See the delete -pcd command.
-------------	---

<b>WARNING</b>	<b>This command may result in data loss. Data should be backed up to other storage before executing this command.</b>
----------------	---



## NOTE

Changing a memory allocation goal modifies how the platform firmware maps persistent memory in the system address space (SPA) which may result in data loss or inaccessible data, but does not explicitly delete or modify user data found in persistent memory.

## OPTIONS

**-f**

**-force**

Reconfiguring DCPMMs is a destructive operation which requires confirmation from the user. This option suppresses the confirmation.

**-h**

**-help**

Displays help for the command.

**-u (B | MB | MiB | GB | GiB | TB | TiB)**

**-units (B | MB | MiB | GB | GiB | TB | TiB)**

Changes the units that capacities are displayed in for this command. One of: bytes (B), megabytes (MB), mebibytes (MiB), gigabytes (GB), gibibytes (GiB), terabytes (TB) or tebibytes (TiB).

## TARGETS

**-dimm [(DimmIDs)]**

Creates a memory allocation goal on specific DCPMMs by optionally supplying one or more comma-separated DCPMM identifiers. This list must include all unconfigured DCPMMs on the affected socket(s). The default is to configure all manageable DCPMMs on all sockets.

**-socket (SocketIds)**

Creates the memory allocation goal onto all manageable DCPMMs on specific sockets by supplying the socket target and one or more comma-separated socket identifiers. The default is to create the memory allocation goal onto all manageable DCPMMs on all sockets.

## PROPERTIES

**MemoryMode**

Percentage of the total capacity to use in Memory Mode (0-100). Default = 0.

**PersistentMemoryType**





If MemoryMode is not 100%, the type of persistent memory to create.

- "AppDirect": (Default) Create App Direct capacity utilizing hardware interleaving across the requested DCPMMs if applicable given the specified target.
- "AppDirectNotInterleaved": Create App Direct capacity that is not interleaved any other DCPMMs.

### NamespaceLabelVersion

The version of the namespace label storage area (LSA) index block

- "1.2": (Default) Defined in UEFI 2.7a - sections 13.19
- "1.1": Legacy 1.1 namespace label support

### Reserved

Reserve a percentage (0-100) of the requested DCPMM App Direct capacity that will not be mapped into the system physical address space and will be presented as Reserved Capacity with Show Device and Show Memory Resources Commands.

## EXAMPLES

Configures all the DCPMM capacity in Memory Mode.

```
ipmctl create -goal MemoryMode=100
```

Configures all the DCPMM capacity as App Direct.

```
ipmctl create -goal PersistentMemoryType=AppDirect
```

Configures the capacity on each DCPMM with 20% of the capacity in Memory Mode and the remaining as App Direct capacity that does not use hardware interleaving.

```
ipmctl create -goal MemoryMode=20  
PersistentMemoryType=AppDirectNotInterleaved
```

Configures the DCPMM capacity across the entire system with 25% of the capacity in Memory Mode, 25% reserved and the remaining 50% as App Direct. Configures the DCPMM capacity across the entire system with 25% of the capacity in Memory Mode and the remaining 75% as App Direct.

```
ipmctl create -goal MemoryMode=25  
PersistentMemoryType=AppDirect Reserved=25
```



## LIMITATIONS

In order to successfully execute this command:

- The caller must have the appropriate privileges.
- The specified DCPMM(s) must be manageable by the host software and must all have the same SKU.
- Existing memory allocation goals that have not been applied and any namespaces associated with the requested DCPMM(s) must be deleted before running this command.
- Security state must be disabled. Changing the memory configuration is a destructive operation which results in loss of data stored in the persistent memory region. Therefore, data should be backed up to other storage before executing this command. Targets may be limited to individual DCPMMs or sockets, but all DCPMMs on affected sockets must be configured when the command finishes. If the selected targets make this impossible, the command will be rejected. Refer to **Show System Capabilities** for a list of BIOS supported modes.
- Some requests are dependent on BIOS and/or platform configuration. For details, refer to the *Intel® Optane™ DC Persistent Memory Software Memory Allocation Rules*, document number 564194. For example:
- Provisioning DCPMMs for Memory Mode while BIOS is configured for 1LM only will result in unused capacity.
- Provisioning DCPMMs for Memory Mode while not all iMCs have at least one DCPMM will result in unused capacity.

## RETURN DATA

Minor adjustments (up to 10%) in the requested capacities are sometimes necessary to align properly according to the platform rules. There are also some situations that require additional confirmation from the user because they may result in a non-optimal configuration (i.e., reduced performance). These are described below:

**The requested goal may result in a non-optimal configuration due to the population of DIMMs in the system**

Memory Mode capacity requested but the population of DRAM DIMMs and DCPMMs in the system may result in reduced performance (i.e., the ratio of DRAM and DCPMMs is not balanced, DRAM and DCPMMs are not on the same channel or not all the same size).

**The requested goal may result in a non-optimal configuration due to the population of DIMMs in the system.**



App Direct capacity requested but the population of DCPMMs in the system may result in reduced performance (i.e., DCPMMs are not the same size or populated asymmetrically across the socket).

**The requested goal will result in App Direct capacity which is not supported by the host software.**

App Direct capacity requested but App Direct is not supported by the currently installed host software.

**The requested goal will result in Memory Mode capacity that is unusable with the currently selected platform BIOS volatile mode.**

Memory Mode capacity requested by the platform BIOS is currently set to 1LM Mode.

**The requested goal was adjusted more than 10% to find a valid configuration.**

> 10% adjustment from the requested goal

**The amount of mapped memory was limited based on the SKU resulting in unmapped capacity.**

Mapped memory was limited based on the CPU SKU.

Therefore, before making any changes to the configuration, a prompt is displayed showing the memory allocation goals that will be created on each DCPMM as documented in the command Section [Show Memory Allocation Goal](#), along with any additional confirmation messages. The force option can be used to override this confirmation and proceed directly with creating the goals.

```
The following configuration will be applied:
SocketID DimmID MemorySize AppDirect1Size
AppDirect2Size (Refer to
the command Section Show Memory Allocation Goal)
[Additional Confirmation Messages (see above)] Do
you want to
continue?
```

## 2.2.2. Show Memory Allocation Goal

### NAME

ipmctl-show-goal - Shows the memory allocation goal on one or more DCPMM

### SYNOPSIS

```
ipmctl show [OPTIONS] -goal [TARGETS] [PROPERTIES]
```



## DESCRIPTION

Shows the memory allocation goal on one or more DCPMMs. Once the goal is successfully applied by the BIOS, it is no longer displayed. Use the command [Show Memory Resources](#) to view the system-wide memory resources or the command **Show Persistent Memory** for detailed persistent memory information.

## OPTIONS

**-a**

**-all**

Shows all attributes.

**NOTE**

The all and display options are exclusive and may not be used together.

**-d (attributes)**

**-display (attributes)**

Filters the returned attributes by explicitly specifying a comma-separated list of any of the attributes defined in the Return Data section.

**NOTE**

The all and display options are exclusive and may not be used together.

**-h**

**-help**

Displays help for the command.

**-u (B | MB | MiB | GB | GiB | TB | TiB)**

**-units (B | MB | MiB | GB | GiB | TB | TiB)**

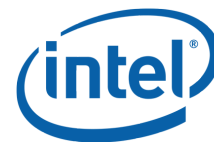
Changes the units that capacities are displayed in for this command. One of: bytes (B), megabytes (MB), mebibytes (MiB), gigabytes (GB), gibibytes (GiB), terabytes (TB) or tebibytes (TiB).

## TARGETS

**-dimm (DimmIDs)**

Restricts output to specific DCPMMs by supplying one or more comma separated DCPMM identifiers. The default is to display all manageable DCPMMs with memory allocation goals.

**-socket (SocketIds)**



Restricts output to the DCPMMs on specific sockets by supplying the socket target and one or more comma-separated socket identifiers. The default is to display all manageable DCPMMs on all sockets with memory allocation goals.

## EXAMPLES

Shows the default memory allocation goal attributes for each DCPMM.

```
ipmctl show -goal
```

Shows all the memory allocation goal attributes for the DCPMMs on socket 1.

```
ipmctl show -a -goal -socket 1
```

## LIMITATIONS

The specified DCPMMs must be manageable by the host software.

## RETURN DATA

The default behavior is to display a table with the default attributes for each DCPMM; applying options changes the output to a more detailed format.

### SocketID

(Default) The processor socket identifier where the DCPMM is installed.

### DimmID

(Default) The DCPMM identifier

### MemorySize

(Default) The DCPMM capacity that will be configured in Memory Mode.

### AppDirect1Size

(Default) The DCPMM capacity that will be configured as the first App Direct interleave set if applicable.

### AppDirect1Index

Unique identifier of the first App Direct interleave set.

- N/A: If no App Direct interleave set
- Numeric value if App Direct interleave set is present.

### AppDirect1Settings

The settings for the first App Direct interleave set in the format: x(Way) [- (Size)



iMC] [x (Size) Channel]

### **AppDirect2Size**

(Default) The DCPMM capacity that will be configured as the second App Direct interleave set if applicable.

### **AppDirect2Index**

Unique identifier of the second App Direct interleave set.

- N/A: If no App Direct interleave set
- Numeric value if App Direct interleave set is present.

### **AppDirect2Settings**

The settings for the second App Direct interleave set in the format: x(Way) [- (Size) iMC] [x (Size) Channel]

### **Status**

The status of the memory allocation goal. One of:

- Unknown: The status cannot be determined.
- New: A reboot is required for the memory allocation goal to be processed by the BIOS.
- Failed - Bad request: The BIOS failed to process the memory allocation goal because it was invalid.
- Failed - Not enough resources: There were not enough resources for the BIOS to process the memory allocation goal.
- Failed - Firmware error: The BIOS failed to process the memory allocation goal due to a firmware error.
- Failed - Unknown: The BIOS failed to process the memory allocation goal due to an unknown error.

### **SAMPLE OUTPUT**

If a new memory allocation goal has been created, a prompt to reboot will be presented.

```
A reboot is required to process new memory
allocation goals.
```



## 2.2.3. Dump Memory Allocation Settings

### NAME

ipmctl-dump-goal - Stores the current system configuration in a file

### SYNOPSIS

```
ipmctl dump [OPTIONS] -destination (path) -system  
-config
```

### DESCRIPTION

Store the currently configured memory allocation settings for all DCPMMs in the system to a file in order to replicate the configuration elsewhere. Apply the stored memory allocation settings using the command Section [Load Memory Allocation Goal](#).

### OPTIONS

**-h**

**-help**

Displays help for the command.

### EXAMPLES

Stores the memory allocation settings from all the DCPMMs into the file "config.txt".

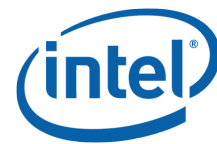
```
ipmctl dump -destination config.txt -system -config
```

### LIMITATIONS

Only memory allocation settings for manageable DCPMMs that have been successfully applied by the BIOS are stored in the file. Unconfigured DCPMMs are not included, nor are memory allocation goals that have not been applied.

### RETURN DATA

The CLI will indicate the overall status of the operation when complete. If a failure occurs when dumping the memory allocation from multiple DCPMMs, the process will stop and the output file will be removed.



The output file is formatted as an ASCII file with one row per DCPMM containing the following comma-separated values.

**SocketID**

unsigned short int Identifier for the socket the DCPMM is associated with.

**DimmHandle**

unsigned int DCPMM device handle.

**Capacity**

unsigned long long int Total capacity of the DCPMM in GiB.

**MemorySize**

unsigned long long int Capacity of the DCPMM allocated as Memory Mode in GiB.

**AppDirect1Size**

unsigned long long int Capacity of the DCPMM allocated for the first App Direct interleave set in GiB.

**AppDirect1Format**

unsigned short int Bit mask representing the interleave format of the first App Direct interleave set.

**AppDirect1Mirrored**

unsigned char 1 if the first App Direct interleave set is mirrored, 0 otherwise.

**AppDirect1Index**

unsigned short int Unique index of the first App Direct interleave set.

**AppDirect2Size**

unsigned long long int Capacity of the DCPMM allocated for the second App Direct interleave set in GiB.

**AppDirect2Format**

unsigned short int Bit mask representing the interleave format of the second App Direct interleave set.

**AppDirect2Mirrored**

unsigned char 1 if the second App Direct interleave set is mirrored, 0 otherwise.

**AppDirect2Index**

unsigned short int Unique index of the second App Direct interleave set.





## SAMPLE OUTPUT

```
Successfully dumped system configuration to file:
config.csv
```

config.csv contents:

```
#SocketID,DimmHandle,Capacity,MemorySize,AppDirect1
Size,AppDirect
1Format,AppDirect
1Mirrored,AppDirect1Index,AppDirect2Size,AppDirect2
Format,AppDire
ct2Mirrored,AppDirect2Index
1,4385,64,64,0,0,0,0,0,0,0,0
1,4401,64,64,0,0,0,0,0,0,0,0
1,4417,64,64,0,0,0,0,0,0,0,0
1,4433,64,64,0,0,0,0,0,0,0,0
1,4449,64,64,0,0,0,0,0,0,0,0
```

## 2.2.4. Load Memory Allocation Goal

### NAME

ipmctl-load-goal - Load a memory allocation goal from a file onto DCPMMs

### SYNOPSIS

```
ipmctl load [OPTIONS] -source (path) -goal
[TARGETS]
```

### DESCRIPTION

Load a memory allocation goal from a file onto one or more DCPMMs.

#### NOTE

Deleting the PCD can be used as a way to prepare individual DCPMMs for provisioning. See the delete -pcd command.

#### WARNING

**This command may result in data loss. Data should be backed up to other storage before executing this command.**



## NOTE

Changing a memory allocation goal modifies how the platform firmware maps persistent memory in the system address space (SPA) which may result in data loss or inaccessible data, but does not explicitly delete or modify user data found in persistent memory.

## OPTIONS

**-f**

**-force**

Reconfiguring DCPMMs is a destructive operation which requires confirmation from the user. This option suppresses the confirmation.

**-h**

**-help**

Displays help for the command.

**-u (B | MB | MiB | GB | GiB | TB | TiB)**

**-units (B | MB | MiB | GB | GiB | TB | TiB)**

Changes the units that capacities are displayed in for this command. One of: bytes (B), megabytes (MB), mebibytes (MiB), gigabytes (GB), gibibytes (GiB), terabytes (TB) or tebibytes (TiB).

## TARGETS

**-dimm (DimmIDs)**

Loads the memory allocation goal onto specific DCPMMs by supplying one or more comma separated DCPMM identifiers. The default is to load the memory allocation goal onto all manageable DCPMMs.

**-socket (SocketIds)**

Loads the memory allocation goal onto all manageable DCPMMs on specific sockets by supplying the socket target and one or more comma-separated socket identifiers. The default is to load the memory allocation goal onto all manageable DCPMMs on all sockets.

## EXAMPLES

Loads the configuration settings stored in "config.txt" onto all the DCPMMs in the system as a memory allocation goal to be applied by the BIOS on the next reboot.

```
ipmctl load -source config.txt -goal
```



Loads the configuration settings stored in "config.txt" onto a specified set of DCPMMs as a memory allocation goal to be applied by the BIOS on the next reboot.

```
ipmctl load -source config.txt -goal -dimm 1,2,3
```

Loads the configuration settings stored in "config.txt" onto all manageable DCPMMs on sockets 1 and 2 as a memory allocation goal to be applied by the BIOS on the next reboot.

```
ipmctl load -source config.txt -goal -socket 1,2
```

## LIMITATIONS

In order to successfully execute this command:

- The caller must have the appropriate privileges.
- The specified DCPMM(s) must be manageable by the host software and must all have the same SKU.
- Existing memory allocation goals that have not been applied and any namespaces associated with the requested DCPMM(s) must be deleted before running this command.
- Security state must be disabled. Changing the memory configuration is a destructive operation which results in loss of data stored in the persistent memory region. Therefore, data should be backed up to other storage before executing this command. Targets may be limited to individual DCPMMs or sockets, but all DCPMMs on affected sockets must be configured when the command finishes. If the selected targets make this impossible, the command will be rejected. Refer to **Show System Capabilities** for a list of BIOS supported modes.
- Some requests are dependent on BIOS and/or platform configuration. For details, refer to the *Intel® Optane™ DC Persistent Memory Software Memory Allocation Rules*, document number 564194. For example:
- Provisioning DCPMMs for Memory Mode while BIOS is configured for 1LM only will result in unused capacity.
- Provisioning DCPMMs for Memory Mode while not all iMCs have at least one DCPMM will result in unused capacity.

## RETURN DATA

If successful, the CLI will display the memory allocation goal stored on each DCPMM as documented in the command Section [Show Memory Allocation Goal](#). If a failure occurs, an error code and message will be displayed. If a failure occurs when



configuring multiple DCPMMs, the process will exit and remove the memory allocation goal from any DCPMMs that succeeded prior to the failure.

## 2.2.5. Delete Memory Allocation Goal

### NAME

ipmctl-delete-goal - Deletes the memory allocation goal from DCPMMs

### SYNOPSIS

```
ipmctl delete [OPTIONS] -goal [TARGETS]
```

### DESCRIPTION

Deletes the memory allocation goal from one or more DCPMMs. This command only deletes a memory allocation goal request that has not been processed by BIOS.

### OPTIONS

**-h**

**-help**

Displays help for the command.

### TARGETS

**-dimm [(DimmIDs)]**

Deletes the memory allocation goal from specific DCPMMs by optionally supplying one or more comma-separated DCPMM identifiers. The default is to delete the memory allocation goals from all manageable DCPMMs.

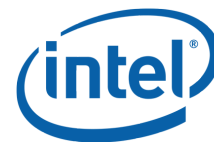
**-socket (SocketIds)**

Deletes the memory allocation goal from the DCPMMs on specific sockets by supplying the socket target and one or more comma-separated socket identifiers. The default is to delete the memory allocation goals from manageable DCPMMs on all sockets.

### EXAMPLES

Deletes the memory allocation goal from all DCPMMs on all sockets.

```
ipmctl delete -goal
```



## LIMITATIONS

The appropriate privileges and the specified DCPMM(s) must be manageable by the host software and unlocked if security is enabled. given socket and all specified DCPMMs must contain a memory allocation goal.

## RETURN DATA

For each DCPMM, the CLI will indicate the status of the operation. If a failure occurs when deleting the memory allocation goals from multiple DCPMMs, the process will output a failure message for those DCPMMs that did not succeed and a success message for those that did.

## SAMPLE OUTPUT

```
Delete memory allocation goal from DIMM (DimmID):  
Success  
Delete memory allocation goal from DIMM (DimmID):  
Error:  
(Description) Delete memory allocation goal from  
DCPMM  
(DimmID): Success  
...
```

## 2.3. Security

DCPMMs support data-at-rest security by encrypting of the data stored in the persistent regions of the DIMM. The CLI supports configuring DCPMM data-at-rest security features including setting a passphrase to lock the DIMM each time it goes through a power cycle, unlocking a DIMM by providing the passphrase, removing the passphrase, preventing security state changes by freeze locking, and crypto-erasing the persistent data by triggering the refresh of the encryption key for the DCPMM persistent region. See the command Enable Device Security for details on setting the data-at-rest security settings.

The CLI supports three mechanisms to specify passphrases in the security commands in order to provide better protection of the passphrase itself.

1. Specify the passphrase(s) directly in the command as properties. This provides the least amount of protection for the passphrase(s) as command line history can be easily retrieved using standard mechanisms.
2. Use empty strings for the passphrase properties (e.g. Passphrase="") in the command which will trigger the CLI to prompt the user for the passphrase(s).



3. Supply a file path to a local file containing the passphrase(s) with the source option, along with setting the passphrase properties to empty strings, and the CLI will read the passphrase(s) from the file.

The file is expected to have the file format on the first line (#ascii is the only supported format) followed by the passphrase properties as described by the individual commands, each on it's own line. An example passphrase file is provided below.

```
#ascii Passphrase=myOldPassphrase123  
NewPassphrase=myNewPassphrase345
```

**NOTE**

It is expected that the same mechanism (property, prompt or file) will be used for all passphrases in a given command

**NOTE**

When specifying a passphrase in the command line, single quotes are not supported and matched double quotes are ignored. (For example, NewPassphrase="123" will result in 123 as the passphrase, but NewPassphrase="123 will result in a syntax error.)

**NOTE**

When specifying a passphrase, the passphrase cannot contain the percent sign (%). Usage of the percent sign in the passphrase will result in a syntax error.

**NOTE**

When specifying a passphrase, double quotes ("), caret (^), pound sign (#), and pipe symbol (|) are stripped out unless escaped with caret. (For example, NewPassphrase=12#34 will result in 1234 as the passphrase, but NewPassphrase=12^#34 will result in 12#34 as the passphrase.)

**NOTE**

When specifying a passphrase in the command line, the passphrase cannot be set to any of the property names associated with the security commands. For example, attempting to use "Passphrase", "NewPassphrase", or "ConfirmPassphrase" as the passphrase will result in a syntax error.

## 2.3.1. Enable Device Security

### NAME

ipmctl-enable-device-security - Enable data-at-rest security on DCPMM



## SYNOPSIS

```
ipmctl set [OPTIONS] -dimm [TARGETS]  
NewPassphrase=(string)  
ConfirmPassphrase=(string)
```

## DESCRIPTION

Enable data-at-rest security for the persistent memory on one or more DCPMMs by setting a passphrase.

## OPTIONS

**-h**

**-help**

Displays help for the command.

**-source (path)**

File path to a local file containing the new passphrase (1-32 characters).

### NOTE

The file does not need to contain the ConfirmPassphrase property

## TARGETS

**-dimm (DimmIDs)**

Set the passphrase on specific DCPMMs by supplying one or more comma separated DCPMM identifiers. However, this is not recommended as it may put the system in an undesirable state. The default is to set the passphrase on all manageable DCPMMs.

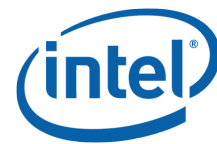
## PROPERTIES

### NewPassphrase

The new passphrase (1-32 characters). For better passphrase protection, specify an empty string (e.g., NewPassphrase="") to be prompted for the passphrase or to use a file containing the passphrase with the source option.

### ConfirmPassphrase

Confirmation of the new passphrase (1-32 character and must match NewPassphrase). For better passphrase protection, specify an empty string (e.g., ConfirmPassphrase="") to be prompted for the passphrase or to use a file containing the passphrase with the source option.



## EXAMPLES

Set a passphrase on DIMM 0x0001.

```
ipmctl set -dimm 0x0001 NewPassphrase=123  
ConfirmPassphrase=123
```

Sets a passphrase on DCPMM 0x0001 by supplying the passphrase in the file mypassphrase.file. In this example, the format of the file would be:

```
#ascii  
NewPassphrase=myNewPassphrase
```

```
ipmctl set -source mypassphrase.file -dimm 0x0001  
NewPassphrase=" " ConfirmPassphrase=" "
```

## LIMITATIONS

In order to successfully execute this command:

The caller must have the appropriate privileges. The specified DCPMM must have security disabled and be manageable by the host software.

There must not be any goal creation pending.

## RETURN DATA

If empty strings are provided for the passphrase properties and the source option is not included, the user will be prompted (once for all DCPMMs) to enter the new passphrase and then again to confirm the new passphrase as described below. The passphrase characters will be hidden.

```
New passphrase: ****  
Confirm new passphrase: ****
```

For each DCPMM, the CLI will indicate the status of the set passphrase operation. If a failure occurs when setting the passphrase on multiple DCPMMs, the process will exit and not continue updating the remaining DCPMMs.

## SAMPLE OUTPUT

```
Set passphrase on DIMM (DimmID): Success  
...  
Set passphrase on DIMM (DimmID): Error (Code) -  
(Description)
```





## 2.3.2. Change Device Passphrase

### NAME

ipmctl-change-device-passphrase - Changes the security passphrase on DCPMM

### SYNOPSIS

```
ipmctl set [OPTIONS] -dimm [TARGETS]  
Passphrase=(string) NewPassphrase=(string)  
ConfirmPassphrase=(string)
```

### DESCRIPTION

Changes the security passphrase on one or more DCPMMs.

### OPTIONS

**-h**

**-help**

Displays help for the command.

**-source (path)**

File path to a local file containing the new passphrase (1-32 characters).

#### NOTE

The file does not need to contain the ConfirmPassphrase property

### TARGETS

**-dimm (DimmIDs)**

Changes the passphrase on specific DCPMMs by supplying one or more comma separated DCPMM identifiers. However, this is not recommended as it may put the system in an undesirable state. The default is to change the passphrase on all manageable DCPMMs.

### PROPERTIES

#### Passphrase

The current passphrase (1-32 characters). For better passphrase protection, specify an empty string (e.g., Passphrase="") to be prompted for the current passphrase or to use a file containing the passphrases with the source option.



## NewPassphrase

The new passphrase (1-32 characters). For better passphrase protection, specify an empty string (e.g., NewPassphrase="") to be prompted for the passphrase or to use a file containing the passphrase with the source option.

## ConfirmPassphrase

Confirmation of the new passphrase (1-32 character and must match NewPassphrase). For better passphrase protection, specify an empty string (e.g., ConfirmPassphrase="") to be prompted for the passphrase or to use a file containing the passphrase with the source option.

## EXAMPLES

Changes the passphrase from mypassphrase to mynewpassphrase on all DCPMMs.

```
ipmctl set -dimm Passphrase=mypassphrase  
NewPassphrase=mynewpassphrase  
ConfirmPassphrase=mynewpassphrase
```

Changes the passphrase on all DCPMMs by having the CLI prompt for the current and new passphrases.

```
ipmctl set -dimm Passphrase=" " NewPassphrase=" "  
ConfirmPassphrase=" "
```

Changes the passphrase on all DCPMMs by supplying the current and new passphrases from the specified file. In this example, the format of the file would be:

```
#ascii  
Passphrase=myOldPassphrase  
NewPassphrase=myNewPassphrase
```

```
ipmctl set -source passphrase.file -dimm  
Passphrase=" " NewPassphrase=" "  
ConfirmPassphrase=" "
```

## LIMITATIONS

The specified DCPMM must be manageable by the host software, have security enabled and not be in the "Unlocked, Frozen", "Disabled, Frozen", or "Exceeded" lock states.



## RETURN DATA

If empty strings are provided for the passphrase properties and the source option is not included, the user will be prompted (once for all DCPMM) to enter the current passphrase, then again for the new passphrase and then again to confirm the new passphrase as described below. The passphrase characters are hidden.

Current passphrase: \*\*\*\*

For each DIMM, the CLI will indicate the status of the passphrase change operation. If a failure occurs when updating the passphrase on multiple DCPMMs, the process will exit and not continue updating the remaining DCPMMs.

## SAMPLE OUTPUT

```
Change passphrase on DIMM (DimmID): Success
...
Change passphrase on DIMM (DimmID): Error (Code)-
(Description)
```

## 2.3.3. Change Device Security

### NAME

ipmctl-change-device-security - Changes the DCPMM security lock state

### SYNOPSIS

```
ipmctl set [OPTIONS] -dimm [TARGETS]
Lockstate=(Unlocked|Disabled|Frozen)
Passphrase=(string)
```

### DESCRIPTION

Changes the data-at-rest security lock state for the persistent memory on one or more DCPMMs.

### OPTIONS

**-h**

**-help**

Displays help for the command.

**-source (path)**

File path to a local file containing the new passphrase (1-32 characters).

**TARGETS****-dimm (DimmIDs)**

Changes the lock state of a specific DCPMMs by supplying one or more comma separated DCPMM identifiers. However, this is not recommended as it may put the system in an undesirable state. The default is to modify all manageable DCPMMs.

**PROPERTIES****LockState**

The desired lock state.

- "Disabled": Removes the passphrase on an DCPMM to disable security. Permitted only when LockState is Unlocked.
- "Unlocked": Unlocks the persistent memory on a locked DCPMM.
- "Frozen": Prevents further lock state changes to the DCPMM until the next reboot.

**Passphrase**

The current passphrase (1-32 characters). For better passphrase protection, specify an empty string (e.g., Passphrase="") to be prompted for the current passphrase or to use a file containing the passphrases with the source option.

**EXAMPLES**

Unlocks device 0x0001.

```
ipmctl set -dimm 0x0001 LockState=Unlocked  
Passphrase= " "
```

Unlocks device 0x0001 by supplying the passphrase in the file "mypassphrase.file". In this example, the format of the file would be:

```
#ascii
```

```
Passphrase=myPassphrase
```

```
ipmctl set -source myfile.file -dimm 0x0001  
LockState=Unlocked  
Passphrase= " "
```



## LIMITATIONS

To successfully execute this command, the caller must have the appropriate privileges and the specified DCPMMs must be manageable by the host software, have security enabled, not be in the "Unlocked, Frozen", "Disabled, Frozen", or "Exceeded" lock states, and not executing a long operation (ARS, Overwrite, FWUpdate).

## RETURN DATA

If an empty string is provided for the passphrase property and the source option is not included, the user will be prompted (once for all DCPMMs) to enter the current passphrase. The passphrase characters are hidden.

Current passphrase: \*\*

For each DCPMM, the CLI will indicate the status of the security state change. If a failure occurs when changing multiple DCPMMs, the process will exit and not continue updating the remaining DCPMMs.

## SAMPLE OUTPUT

```
Unlock DIMM (DimmID): Success
Unlock DIMM (DimmID): Error (Code) - (Description)
Remove passphrase from DIMM (DimmID): Success
Remove passphrase from DIMM (DimmID): Error (Code)
- (Description)
```

# 2.3.4. Erase Device Data

## NAME

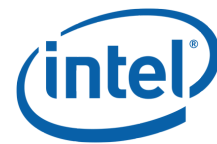
ipmctl-erase-device-data - Erases the persistent data on one or more DCPMMs

## SYNOPSIS

```
ipmctl delete [OPTIONS] -dimm [TARGETS]
Passphrase=(string)
```

## DESCRIPTION

Erases the persistent data on one or more DCPMMs.



## OPTIONS

**-f**

**-force**

Erasing DCPMM data is a destructive operation which requires confirmation from the user for each DCPMM. This option suppresses the confirmation.

**-h**

**-help**

Displays help for the command.

**-source (path)**

File path to a local file containing the new passphrase (1-32 characters).

## TARGETS

**-dimm (DimmIDs)**

Erases specific specifics by supplying one or more comma-separated specific identifiers. However, this is not recommended as it may put the system in an undesirable state. The default is to erase all manageable DCPMMs.

## PROPERTIES

### Passphrase

If security state is disabled, then passphrase is not required and will be ignored if supplied.

If security state is enabled, then a passphrase must be supplied.

The current passphrase (1-32 characters). For better passphrase protection, specify an empty string (e.g., `Passphrase=""`) to be prompted for the passphrase or to use a file containing the passphrase with the source option.

## EXAMPLES

Security disabled DCPMMs: Erases all the persistent data on all DCPMMs in the system.

```
ipmctl delete -dimm
```

Security enabled specifics: Erases all the persistent data on all DCPMMs in the system.

```
ipmctl delete -dimm Passphrase=123
```

Erases all the persistent data on all DCPMMs by having the CLI prompt for the current passphrase.



```
ipmctl delete -dimm Passphrase=" "
```

## LIMITATIONS

To successfully execute this command, the caller must have the appropriate privileges and the specified DCPMM(s) must be manageable by the host software, not be in the "Unlocked, Frozen", "Disabled, Frozen", or "Exceeded" lock states and any namespaces associated with the requested DCPMM(s) must be deleted before running this command.

## RETURN DATA

If an empty string is provided for the passphrase property and the source option is not included, the user will be prompted (once for all DCPMMs) to enter the current passphrase. The passphrase characters are hidden.

Current passphrase: \*\*

For each DCPMM, the CLI will indicate the status of the security state change. If a failure occurs when changing multiple DCPMMs, the process will exit and not continue updating the remaining DCPMMs.

## SAMPLE OUTPUT

```
Erase DIMM (DimmID): Success
Erase DIMM (DimmID): Error (Code) - (Description)
```

# 2.4. Instrumentation

## 2.4.1. Show Sensor

### NAME

ipmctl-show-sensor - Shows health statistics for one or more DCPMMs

### SYNOPSIS

```
ipmctl show [OPTIONS] -sensor [SENSORS] [TARGETS]
```

### DESCRIPTION

ipmctl-show-device to view more detailed information about a DCPMM.



## OPTIONS

**-a**

**-all**

Shows all attributes.

**NOTE**

The all and display options are exclusive and may not be used together.

**-d (attributes)**

**-display (attributes)**

Filters the returned attributes by explicitly specifying a comma-separated list of any of the attributes defined in the Return Data section.

**NOTE**

The all and display options are exclusive and may not be used together.

**-h**

**-help**

Displays help for the command.

## SENSORS

### Health

(Default) The current DCPMM health as reported in the SMART log

### MediaTemperature

(Default) The current DCPMM media temperature in Celsius

### ControllerTemperature

(Default) The current DCPMM controller temperature in Celsius

### PercentageRemaining

(Default) Remaining DCPMM's life as a percentage value of factory expected life span

### LatchedDirtyShutdownCount

(Default) The number of shutdowns without notification over the lifetime of the DCPMM

### UnlatchedDirtyShutdownCount

(Default) The number of shutdowns without notification over the lifetime of the





DCPMM. This counter is the same as LatchedDirtyShutdownCount except it will always be incremented on a dirty shutdown, even if Latch System Shutdown Status was not enabled

#### **PowerOnTime**

(Default) The total power-on time over the lifetime of the DCPMM

#### **UpTime**

(Default) The total power-on time since the last power cycle of the DCPMM

#### **PowerCycles**

(Default) The number of power cycles over the lifetime of the DCPMM

#### **FwErrorCount**

The total number of firmware error log entries

### **TARGETS**

#### **-dimm (DimmIDs)**

Restricts output to the sensors on specific DCPMMs by supplying the DIMM target and one or more comma-separated DCPMM identifiers. The default is to display sensors for all manageable DCPMMs.

### **EXAMPLES**

Gets all sensor information for all DCPMMs.

```
ipmctl show -sensor
```

Shows the media temperature sensor for the specified DCPMM.

```
ipmctl show -sensor MediaTemperature -dimm 1234
```

### **LIMITATIONS**

The specified DCPMM(s) must be manageable by the host software.

### **RETURN DATA**

This command displays a table with a row for each sensor on each DCPMM. Applying a specific DCPMM or sensor type target limits the number of rows. Applying options can be used to expand or restrict the output.

#### **DimmID**



(Default) The DCPMM identifier

**Type**

(Default) The sensor type. Refer to the sensor table above.

**CurrentValue**

(Default) The current reading followed by the units of measurement (e.g., 57 °C or 25%)

**CurrentState**

(Default) The current value in relation to the threshold settings (if supported). One of:

- Unknown: The state cannot be determined.
- Normal: The current reading is within the normal range. This is the default when the sensor does not support thresholds.
- NonCritical: The current reading is within the non-critical range. For example, an alarm threshold has been reached.
- Critical: The current reading is within the critical range. For example, the firmware has begun throttling down traffic to the DCPMM due to the temperature.
- Fatal: The current reading is within the fatal range. For example, the firmware is shutting down the DCPMM due to the temperature.

**LowerThresholdNonCritical**

The threshold value below which the state is considered "NonCritical".

**UpperThresholdNonCritical**

The threshold value at or above which the state is considered "NonCritical".

**LowerThresholdCritical**

The threshold value below which the state is considered "Critical".

**UpperThresholdCritical**

The threshold value at or above which the state is considered "Critical".

**UpperThresholdFatal**

The threshold value at or above which the state is considered "Fatal".

**SettableThresholds**

A list of user settable thresholds. Zero or more of:



- "LowerThresholdNonCritical"
- "UpperThresholdNonCritical"

### SupportedThresholds

A list of supported thresholds. Zero or more of:

- "LowerThresholdNonCritical"
- "UpperThresholdNonCritical"
- "LowerThresholdCritical"
- "UpperThresholdCritical"
- "UpperThresholdFatal"

### EnabledState

Whether the critical threshold alarm is enabled, disabled or not applicable. One of:

- 0: Disabled
- 1: Enabled
- N/A

## 2.4.2. Change Sensor Settings

### NAME

ipmctl-change-sensor - Changes the threshold or enabled state for DCPMMs sensors

### SYNOPSIS

```
ipmctl set [OPTIONS] -sensor (SENSORS) [TARGETS]  
NonCriticalThreshold=(temperature)  
EnabledState=(0|1)
```

### DESCRIPTION

Changes the non-critical threshold or enabled state for one or more DCPMMs sensors. Use the command Show Sensor to view the current settings.

### OPTIONS

- f
- force



Changing the sensor settings is a potentially destructive operation which requires confirmation from the user for each DCPMM. This option suppresses the confirmation.

**-h**

**-help**

Displays help for the command.

## SENSORS

### MediaTemperature

The current DCPMM media temperature in Celsius.

Valid values: 0-2047

### ControllerTemperature

The current DCPMM controller temperature in Celsius.

Valid values 0-2047

### PercentageRemaining

Remaining DCPMM's life as a percentage value of factory expected life span.

Valid values: 1-99

## TARGETS

### -dimm [(DimmIDs)]

Restricts output to the sensors on specific DCPMMs by optionally supplying the DIMM target and one or more comma-separated DCPMM identifiers. The default is to display sensors for all manageable DCPMMs.

## PROPERTIES

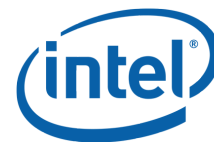
### NonCriticalThreshold

The upper (for temperatures) or lower (for spare capacity) non-critical alarm threshold of the sensor. If the current value of the sensor is at or above for thermal, or below for capacity, the threshold value, then the sensor will indicate a "NonCritical" state. Temperatures may be specified in degrees Celsius to a precision of 1/16 a degree.

### EnabledState

Enable or disable the non-critical threshold alarm. One of:

- "0": Disable
- "1": Enable



## EXAMPLES

Changes the media temperature threshold to 51 on the specified DCPMM and enable the alarm.

```
ipmctl set -sensor MediaTemperature -dimm 0x0001
NonCriticalThreshold=51
EnabledState=1
```

## LIMITATIONS

The specified DCPMM(s) must be manageable by the host software.

## RETURN DATA

For each DCPMM, the CLI will indicate the status of the operation. If a failure occurs when modifying multiple DCPMMs, the process will exit and not continue modifying the remaining DCPMMs.

## SAMPLE OUTPUT

```
Modify (Sensor) settings on DIMM (DimmID): Success
...
Modify (Sensor) settings on DIMM (DimmID): Error
(Code) -
(Description)
```

# 2.5. Persistent Memory Provisioning

During the initial provisioning of the Section [\[Memory Subsystem Provisioning\]](#) it is possible to allocate some or all of the available DCPMM capacity as persistent memory. If this was done, the result is one or more regions of persistent memory. The number of persistent memory regions depends on the platform configuration (e.g., is it a 2 socket or a 4 socket server) and the type of initial provisioning request.

Persistent memory provisioning is the process by which the capacity in available regions is partitioned and made available as a persistent memory device or "namespace" in the file system. Due to a number of factors there may be restrictions on the type, size and number of namespaces that can be created from a given region. The output of the command [Show Persistent Memory](#) includes information on about the persistent memory regions. FreeCapacity indicates the largest that can be created



from a given region. A size of zero indicates that it's not possible to create that namespace type from the region whose output is being examined.

See the command Section [Create Namespace](#) below for more information on how to allocate namespaces from regions. Once namespaces have been created information about them can be displayed with the command Section [Show Namespace](#) Once a given namespace is no longer needed it can be deleted with the command Section [Delete Namespace](#) When a namespace is deleted its capacity is returned to the region from which it was created and it can be reallocated as a new namespace.

<b>NOTE</b>	Deleting a namespace modifies how the platform organizes the persistent memory, but does not explicitly delete or modify user data found in persistent memory
<b>NOTE</b>	If DCPMM is locked, then namespaces cannot be detected or displayed.
<b>NOTE</b>	Namespace management is not supported via ipmctl CLI in the Operating System (OS) environment. Native OS utilities must be used for namespace management. See the Intel® Optane™ DC Persistent Memory Help and User's Guide (document number 576369) for additional details on using the native OS utilities.

## 2.5.1. Show Persistent Memory

### NAME

ipmctl-show-region - Retrieves a list of persistent memory regions

### SYNOPSIS

```
ipmctl show [OPTIONS] -region [TARGETS]
```

### DESCRIPTION

Retrieves a list of persistent memory regions of DCPMM capacity

### OPTIONS

**-a**

**-all**

Shows all attributes.

**NOTE**

The all and display options are exclusive and may not be used together.

**-d (attributes)****-display (attributes)**

Filters the returned attributes by explicitly specifying a comma-separated list of any of the attributes defined in the Return Data section.

**NOTE**

The all and display options are exclusive and may not be used together.

**-h****-help**

Displays help for the command.

**-u (B|MB|MiB|GB|GiB|TB|TiB)****-units (B|MB|MiB|GB|GiB|TB|TiB)**

Changes the units that capacities are displayed in for this command. One of: bytes (B), megabytes (MB), mebibytes (MiB), gigabytes (GB), gibibytes (GiB), terabytes (TB) or tebibytes (TiB).

## TARGETS

**-region (RegionIDs)**

Restricts output to specific persistent memory regions by providing one or more comma separated region identifiers. The default is to display the persistent memory regions across all manageable DCPMMs.

**-socket (SocketIDs)**

Restricts output to the persistent memory regions on specific sockets by supplying the socket target and one or more comma-separated socket identifiers. The default is to display all sockets.

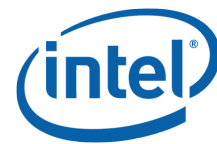
## EXAMPLES

Shows all attributes of all persistent memory regions in the server.

```
ipmctl show -a -region
```

Shows all attributes for the specified persistent memory region.

```
ipmctl show -a -region 1
```



## LIMITATIONS

All the underlying DCPMMs should be unlocked to accurately reflect the available capacities. The specified DCPMM(s) must be manageable by the host software.

## RETURN DATA

The default behavior is to display a table with the default attributes listed below; applying options changes the output to a more detailed format.

### **RegionID**

(Default) The unique region identifier

### **PersistentMemoryType**

(Default) A comma-separated list of the underlying type(s) of persistent memory capacity in the region. One or more of:

- AppDirect: App Direct capacity interleaved across two or more DCPMMs that is fully mapped into the system physical address space.
- AppDirectNotInterleaved: App Direct capacity wholly contained on a single DCPMMs that is fully mapped into the system physical address space.

### **Capacity**

(Default) Total usable capacity, both allocated and unallocated

### **FreeCapacity**

(Default) Remaining usable capacity

### **SocketID**

(Default) Socket ID to which the region belongs

### **HealthState**

The rolled up health of the underlying DCPMMs. One of:

- Unknown: The region health cannot be determined.
- Healthy: All underlying DCPMM persistent memory capacity is available.
- Pending: A new memory allocation goal has been created but not applied. Reboot or delete any existing memory allocation goals before creating namespaces on the region.
- Error: There is an issue with some or all of the underlying DCPMM capacity because the interleaved set has failed.
- Locked: One or more of the of the underlying DCPMMs are locked.





### **ISetID**

The region unique identifier. Also known as interleave set cookie.

### **DimmID**

A list of all the DIMMs that are part of this reg.

## **2.5.2. Show Namespace**

### **NAME**

ipmctl-show-namespace - Shows information about one or more namespaces

### **SYNOPSIS**

```
ipmctl show [OPTIONS] -namespace [TARGETS]
```

### **DESCRIPTION**

Shows information about one or more namespaces.

### **OPTIONS**

**-a**

**-all**

Shows all attributes.

#### **NOTE**

The all and display options are exclusive and may not be used together.

**-d (attributes)**

**-display (attributes)**

Filters the returned attributes by explicitly specifying a comma-separated list of any of the attributes defined in the Return Data section.

#### **NOTE**

The all and display options are exclusive and may not be used together.

**-h**

**-help**

Displays help for the command.



**-u (B | MB | MiB | GB | GiB | TB | TiB)**

**-units (B | MB | MiB | GB | GiB | TB | TiB)**

Changes the units that capacities are displayed in for this command. One of: bytes (B), megabytes (MB), mebibytes (MiB), gigabytes (GB), gibibytes (GiB), terabytes (TB) or tebibytes (TiB).

## TARGETS

**-namespace (NamespaceIDs)**

Restricts output to specific namespaces by providing a comma-separated list of one or more namespace identifiers. The default is to display all namespaces.

**-region (RegionIDs)**

Restricts output to the namespaces on specific regions by supplying the region target and one or more comma-separated region identifiers. The default is to display namespaces on all regions.

## EXAMPLES

Shows the default attributes for the specified namespace.

```
ipmctl show -namespace 0x01
```

Shows all the attributes for all namespaces.

```
ipmctl show -a -namespace
```

## LIMITATIONS

Namespaces are not reported from locked DCPMMs. Use the command **Show System Capabilities** to retrieve namespace limitations as reported by the driver.

## RETURN DATA

The default behavior is to display a table with the default attributes listed below; applying options changes the output to a more detailed format.

### NamespaceID

(Default) The identifier of the namespace

### Capacity

(Default) The usable capacity of the namespace. This is raw namespace capacity less any meta data.

**HealthState**

(Default) The rolled-up health of the underlying DCPMMs. One of:

- Unknown
- Healthy
- Warning: Indicates a namespace has a non-critical problem: reporting an LBA size that is unsupported by this driver.
- Critical: Indicates a namespace has a critical problem: inconsistent data structures, mismatch between namespace label and associated PM region, poison or uncorrectable data in namespace metadata.
- Locked: One or more of the of the underlying DCPMMs are locked.
- Unsupported: The namespace type is not supported.

**Name**

The user specified namespace name up to a maximum of 64 characters

**RegionID**

The region from which the namespace was created.

**BlockSize**

The logical size in bytes for compatibility with block read/write operations

**Mode**

Address abstraction type found on the namespace. One of:

- None (raw)
- Sector - Powerfail write atomicity is guaranteed via a Block Translation Table (BTT).
- fsdax - Supports filesystem-dax (See ndctl-create-namespace)

**NOTE** | If the namespace is disabled, mode may be reported as "None".

**LabelVersion**

Indicates the Namespace index version that can be set with Create Memory Allocation Goal. One of:

- 1.1
- 1.2

**NamespaceUUID**



UUID of the namespace (raw UUID)

## 2.5.3. Create Namespace

### NAME

`ipmctl-create-namespace` - Creates a namespace from a persistent memory region

### SYNOPSIS

```
ipmctl create [OPTIONS] -namespace -region  
(RegionID) [PROPERTIES]
```

### DESCRIPTION

Creates a new namespace from a persistent memory region of DCPMM capacity.

### OPTIONS

**-f**

**-force**

If the resulting namespace capacity does not align properly, a prompt will be displayed to confirm the rounded-up namespace capacity to the next aligned size. This option suppresses the confirmation and proceeds with rounding up the namespace capacity as necessary.

**-h**

**-help**

Displays help for the command.

**-u (B|MB|MiB|GB|GiB|TB|TiB)**

**-units (B|MB|MiB|GB|GiB|TB|TiB)**

Changes the units that capacities are displayed in for this command. One of: bytes (B), megabytes (MB), mebibytes (MiB), gigabytes (GB), gibibytes (GiB), terabytes (TB) or tebibytes (TiB).

### TARGETS

**-region (RegionIDs)**

The region identifier on which to create the namespace



## PROPERTIES

### Capacity

The size of the namespace. Unless the units option is provided, capacity is expected in GiB. The default is to use the maximum available capacity on the specified region.

### Name

Optional user specified namespace name to more easily identify the namespace. Up to a maximum of 63 characters.

### Mode

The mode for the namespace after creation. One of:

- "None": (Default) Raw namespace without any address abstraction.
- "Sector": Powerfail block write atomicity is guaranteed via a Block Translation Table (BTT)

## EXAMPLES

Creates an App Direct memory namespace on region 0x01 using default settings.

```
ipmctl create -namespace -region 0x01
```

Creates a 32 GB App Direct namespace on region 0x01.

```
ipmctl create -units GB -namespace -region 0x01  
Capacity=32
```

Creates a 32 GB App Direct sector mode namespace on region 0x01.

```
ipmctl create -units GB -namespace -region 0x01  
Capacity=32 mode=Sector
```

## LIMITATIONS

To successfully execute this command, the underlying DCPMMs must be unlocked and the region HealthState must be "Healthy."

Uses the command **Show System Capabilities** to retrieve namespace limitations as reported by the driver.

**NOTE**

For App Direct namespaces to be compatible across operating systems, it is recommended that a single App Direct namespace consumes the entire capacity (AppDirectNamespaceMaxSize) as reported by the command **Show Persistent Memory**.

**NOTE**

Microsoft\* Windows\* validates only a single namespace per region configuration.

**RETURN DATA**

If the resulting namespace capacity does not align properly, a prompt will be displayed to confirm the rounded-up namespace capacity to the next aligned size. The force option can be used to override this confirmation and proceed with the rounded up capacity.

```
The requested namespace capacity [capacity] will be
rounded up to [capacity] to
align properly. Do you want to continue?
```

**SAMPLE OUTPUT**

If the namespace is created successfully, the CLI will display all attributes of the new namespace as documented in the command Section [Show Namespace](#). If a failure occurs when creating the namespace, the CLI will display a single error message.

```
Create namespace failed: Error (Code)-
(Description)
```

## 2.5.4. Modify Namespace

**NAME**

ipmctl-modify-namespace - Modifies one or more existing namespaces

**SYNOPSIS**

```
ipmctl set [OPTIONS] -namespace (NamespaceIDs)
[PROPERTIES]
```

**DESCRIPTION**

Modifies one or more existing namespaces.



## OPTIONS

**-f**

**-force**

Modifying namespaces is a potentially destructive operation which requires confirmation from the user for each namespace. This option suppresses the confirmation.

**-h**

**-help**

Displays help for the command.

**-u (B | MB | MiB | GB | GiB | TB | TiB)**

**-units (B | MB | MiB | GB | GiB | TB | TiB)**

Changes the units that capacities are displayed in for this command. One of: bytes (B), megabytes (MB), mebibytes (MiB), gigabytes (GB), gibibytes (GiB), terabytes (TB) or tebibytes (TiB).

## TARGETS

**-namespace (NamespaceIDs)**

Modifies the settings on specific namespaces by providing comma separated list of one or more namespace identifiers. The default is to modify all namespaces.

## PROPERTIES

### Capacity

Changes the size of the namespace. Unless the units option is provided, capacity is expected in GiB.

#### NOTE

Expanding a namespace is only possible if there is available capacity on the region from which the namespace was created and the namespace is not in a critical health state. Refer to the command **Show System Capabilities** to determine if the host software supports changing the size of a namespace.

### Name

Changes the user specified namespace name up to a maximum of 64 characters. Refer to the command **Show System Capabilities** to determine if the host software supports changing the name of a namespace.



## EXAMPLES

Changes the name of the namespace to "MyNewNamespaceName"

```
ipmctl set -namespace 12345678-1234-1234-1234-  
012345678912 Name=MyNewNamespaceName
```

## LIMITATIONS

To successfully execute this command, the underlying DCPMMs must be unlocked.

The capacity cannot be modified for namespaces with Sector optimization enabled.

Uses the command **Show System Capabilities** to retrieve namespace limitations as reported by the driver.

## RETURN DATA

If the resulting namespace capacity does not align properly, a prompt will be displayed to confirm the adjusted namespace capacity. The force option can be used to override this confirmation and proceed with the adjusted capacity.

```
The requested namespace capacity [capacity] will be  
changed to  
[capacity] to align properly. Do you want to  
continue?
```

For each namespace, the CLI will indicate the status of the operation. If a failure occurs when modifying multiple namespaces, the process will continue modifying the remaining namespaces. Use the command Section [Show Namespace](#) to view the modified namespace(s)

## SAMPLE OUTPUT

```
Modify namespace (NamespaceID): Success  
...  
Modify namespace (NamespaceID): Success  
Modify namespace (NamespaceID): Error (Code) -  
(Description)  
[Failed to restore the original state.]
```

## 2.5.5. Delete Namespace





## NAME

ipmctl-delete-namespace - Deletes one or more existing namespaces

## SYNOPSIS

```
ipmctl delete [OPTIONS] -namespace (NamespaceIDs)
```

## DESCRIPTION

Deletes one or more existing namespaces. All data on the deleted namespace(s) becomes inaccessible.

### NOTE

Changing the namespace configuration is a destructive operation which may result in loss of data stored in the associated namespace. Therefore, data should be backed up to other storage before executing this command.

### NOTE

Deleting a namespace removes the logical OS mapping to the persistent memory data, but does not explicitly delete or modify user data found in persistent memory.

## OPTIONS

**-f**

**-force**

Deleting namespaces is a destructive operation which requires confirmation from the user for each namespace. This option suppresses the confirmation.

**-h**

**-help**

Displays help for the command.

## TARGETS

**-namespace (NamespaceIDs)**

Deletes specific namespaces by providing a comma-separated list of one or more namespace identifiers. The default is to delete all namespaces. Namespace identifiers are limited to 64 characters.

## EXAMPLES

Deletes namespace 0x02.



```
ipmctl delete -namespace 0x2
```

## LIMITATIONS

To successfully execute this command, any associated file systems must be unmounted.

The specified DCPMM(s) must be manageable by the host software and unlocked if security is enabled.

## RETURN DATA

For each namespace, the CLI will indicate the status of the operation. If a failure occurs when deleting multiple namespaces, the process will continue deleting the remaining namespaces.

## SAMPLE OUTPUT

```
Delete namespace (NamespaceID): Success
Delete namespace (NamespaceID Error (Code) -
(Description) Delete namespace (NamespaceID):
Success
...
```

# 2.6. Support and Maintenance

## 2.6.1. Help

### NAME

ipmctl-help - Shows help for the supported commands

### SYNOPSIS

```
ipmctl help [OPTIONS]
```

### DESCRIPTION

Shows help for the supported commands.



## OPTIONS

**-h**

**-help**

Displays help for the command.

## EXAMPLES

Lists all supported commands

```
ipmctl help
```

## RETURN DATA

The default behavior is to display an introduction to CLI followed by a list of the supported commands. To display detailed help for a specific command, use the help option with that specific command.

## SAMPLE OUTPUT

```
ipmctl help
Commands:
(name)
(synopsis)
(name)
(synopsis)
(name)
(synopsis)
(name)
(synopsis)
...
```

## 2.6.2. Version

### NAME

ipmctl-version - Shows the DCPMM host software versions

### SYNOPSIS

```
ipmctl version [OPTIONS]
```



## DESCRIPTION

Shows the DCPMM host software versions.

## OPTIONS

**-h**

**-help**

Displays help for the command.

## EXAMPLES

Displays the available version information for the DCPMM software components.

```
ipmctl version
```

## RETURN DATA

By default returns the following inventory information.

### Component

The name of the software component. One of:

- [Product Name] Software Version: The DCPMM management software version
- [Product Name] Driver Version: The vendor specific DCPMM driver version

### Version

The current version of the software component if found or an error if not.

### NOTE

If the software version is incompatible, the version will be followed by an error message indicating such. If DCPMMs are found with a FIS implementation higher than supported by the SW version, this command will print a warning.

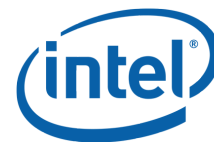
## 2.6.3. Show Device Firmware

### NAME

ipmctl-show-firmware - Shows detailed information about the firmware

### SYNOPSIS

```
ipmctl show [OPTIONS] -firmware [TARGETS]
```



## DESCRIPTION

Shows detailed information about the firmware on one or more DCPMMs.

## OPTIONS

**-a**

**-all**

Shows all attributes.

### NOTE

The all and display options are exclusive and may not be used together.

**-d (attributes)**

**-display (attributes)**

Filters the returned attributes by explicitly specifying a comma-separated list of any of the attributes defined in the Return Data section.

### NOTE

The all and display options are exclusive and may not be used together.

**-h**

**-help**

Displays help for the command.

## TARGETS

**-dimm (DimmIDs)**

Restricts output to the firmware information for specific DCPMMs by supplying one or more comma-separated DCPMM identifiers. The default is to display the firmware information for all manageable DCPMMs.

## EXAMPLES

Shows the firmware information for all DCPMMs in the server.

```
ipmctl show -dimm -firmware
```

## LIMITATIONS

The specified DCPMM(s) must be manageable by the host software.



## RETURN DATA

The default behavior is to display a table with the default attributes listed below; the options can be used to expand or restrict the output.

### **DimmID**

(Default) The DCPMM identifier

### **ActiveFWVersion**

(Default) The BCD-formatted revision of the active firmware in the format PN.RN.SV.bbbb where:

- PN = 2-digit product number
- RN = 2-digit revision number
- SV = 2-digit security version number
- bbbb = 4-digit build version

### **StagedFWVersion**

(Default) The BCD-formatted revision of the firmware staged for execution on the next power cycle in the format PN.RN.SV.bbbb where:

- PN = 2-digit product number
- RN = 2-digit revision number
- SV = 2-digit security version number
- bbbb = 4-digit build version

### **FWUpdateStatus**

The status of the last firmware update operation. One of:

- Unknown
- Staged successfully
- Update loaded successfully
- Update failed to load, fell back to previous firmware

### **FWImageMaxSize**

The maximum size of a firmware image

## 2.6.4. Update Firmware



## NAME

ipmctl-update-firmware - Updates the firmware on one or more DCPMMs

## SYNOPSIS

```
ipmctl load [OPTIONS] -source (path) -dimm  
(DimmIds) [TARGETS]
```

## DESCRIPTION

Updates the firmware on one or more DCPMMs. On the next power cycle, the firmware will become active.

### NOTE

If Address Range Scrub (ARS) is in progress on any target DIMM, an attempt will be made to abort ARS and the proceed with the firmware update.

### NOTE

A reboot is required to activate the updated firmware image and is recommended to ensure ARS runs to completion.

## OPTIONS

**-x**

**-examine**

Verifies the target DIMM is compatible with and ready to receive the recovery firmware image specified in the source option. Return the recovery firmware image version.

**-f**

**-force**

Downgrading the firmware to an older version is a potentially destructive operation which requires confirmation from the user for each DCPMM. This option suppresses the confirmation when attempting to downgrade.

**-h**

**-help**

Displays help for the command.

**-recover**

The recover option forces the firmware update to attempt using SMBus instead of DDRT. This is necessary in case the DDRT link is not trained. This update method will be much slower than using the DDRT interface.



### **-recover FlashSPI**

The FlashSPI option is a modifier that forces an update of the entire SPI Flash. This requires a specific flash image that is intended as full SPI flash update. This update method is only possible if existing SPI firmware images are all corrupt and cannot be loaded by the DIMM's ROM.

## **TARGETS**

### **-dimm (DimmIDs)**

Updates the firmware on specific DCPMMs by supplying one or more comma-separated DCPMM identifiers. However, this is not recommended as it may put the system in an undesirable state. The default is to update all manageable DCPMMs.

## **EXAMPLES**

Updates the firmware on all DCPMMs in the system to the image in sourcefile.pkg on the next power cycle.

```
ipmctl load -source sourcefile.pkg -dimm
```

Checks the firmware image in c:\sourcefile.pkg and retrieve the version.

```
ipmctl load -examine -source sourcefile.pkg -dimm
```

## **LIMITATIONS**

The specified DCPMM(s) must be manageable by the host software.

Firmware version (PN.RN.SV.bbbb) updates are supported as follows:

- The product number (PN) cannot be changed.
- The revision number (RN) can be upgraded when PN is the same.
- The security revision number (SV) can be upgraded when PN.RN is the same. In some configurations it can also be downgraded when PN.RN is the same; use the examine option to determine if the security revision number can be downgraded.
- The build number (bbbb) can be upgraded or downgraded. However, if the firmware API version in the firmware image is lower than is supported by the host software and would make the DCPMM become unmanageable, the downgrade is not supported.



**NOTE**

Once a firmware image is staged for execution, a power cycle is required before another firmware image of the same type (production or debug) can be staged for execution using this command.

**RETURN DATA**

When the examine option is provided, the firmware image is checked and the version number and firmware type is provided. The firmware will either be valid for the DCPMM, a downgrade or invalid meaning it cannot be used for that DCPMM.

**SAMPLE OUTPUT**

```
(file path): MM.mm.hh.bbbb
Load FW on DIMM (DimmID): (Valid|Downgrade) [(with
confirmation or the force option)]
```

If the firmware is being downgraded and the force option is not provided, the user will be prompted to confirm the downgrade for each DCPMM. Otherwise, for each DCPMM, the CLI will indicate the status of the operation.

```
Downgrade firmware on DIMM (DimmID)? (y or [n])
Downgrade firmware
on DIMM (DimmID)? (y or [n])
...
```

If a failure occurs when updating multiple DCPMMs, the process will continue attempting to update the remaining DCPMMs requested. The firmware will not become active until the next power cycle. Use the command [Section Show Device Firmware](#) to view more detailed information about the active and staged firmware.

```
Load FW on DIMM (DimmID): Success, a power cycle is
required to
activate the FW.
Load FW on DIMM (DimmID): Success, a power cycle is
required to
activate the FW.
...
Load FW on DIMM (DimmID): Error (Code) -
(Description)
```



## 2.6.5. Show Preferences

### NAME

ipmctl-show-preferences - Displays a list of the DCPMM software user preferences

### SYNOPSIS

```
ipmctl show [OPTIONS] -preferences
```

### DESCRIPTION

Displays a list of the DCPMM software user preferences and their current values.

### OPTIONS

**-h**

**-help**

Displays help for the command.

### EXAMPLES

Displays the current values for all the user preferences

```
ipmctl show -preferences
```

### RETURN DATA

#### CLI\_DEFAULT\_DIMM\_ID

The default display of DCPMM identifiers. One of:

- **UID:** Use the DimmUID attribute as defined in the command Show Device.
- **HANDLE:** Use the DimmHandle attribute as defined in the command Show Device. This is the default.

#### CLI\_DEFAULT\_SIZE

The default display of capacities in the CLI. One of:

- **AUTO:** Automatically choose the best format for each capacity in binary multiples of bytes (i.e., B, MiB, GiB or TiB). This is the default.
- **AUTO\_10:** AUTO\_10: Automatically choose the best format for each capacity in decimal multiples of bytes (i.e., B, MB, GB or TB).



- B: Displays all capacities in bytes.
- MB: Displays all capacities in megabytes.
- MiB: Displays all capacities in mebibytes.
- GB: Displays all capacities in gigabytes.
- GiB: Displays all capacities in gibibytes.
- TB: Displays all capacities in terabytes.
- TiB: Displays all capacities in tebibytes.

#### **PERFORMANCE\_MONITOR\_ENABLED**

Whether or not the monitor is periodically storing performance metrics for the DCPMMs in the host server. One of:

- 0: Disabled
- 1: Enabled. This is the default.

#### **PERFORMANCE\_MONITOR\_INTERVAL\_MINUTES**

The interval in minutes that the monitor is retrieving and storing performance metrics (if enabled). The default value is 180 minutes.

#### **EVENT\_MONITOR\_ENABLED**

Whether or not the monitor is periodically checking for DCPMM events. One of:

- 0: Disabled
- 1: Enabled. This is the default.

#### **EVENT\_MONITOR\_INTERVAL\_MINUTES**

The interval in minutes that the monitor is checking for and storing DCPMM events (if enabled). The default value is 1 minute.

#### **EVENT\_LOG\_MAX**

The maximum number of events to keep in the management software. The default value is 0, which indicates no limit.

#### **DBG\_LOG\_MAX**

The maximum number of debug log entries to keep in the management software. The default value is 0, which indicates no limit.

#### **APPDIRECT\_SETTINGS**

The interleave settings to use when creating App Direct capacity in the format: (IMCSize\_ChannelSize). The default is "RECOMMENDED" which uses the BIOS



recommended App Direct settings returned by the command Show System Capabilities.

#### **APPDIRECT\_GRANULARITY**

The minimum App Direct granularity per DCPMM supported by the command Create Memory Allocation Goal. One of:

- RECOMMENDED: Use the recommended App Direct granularity of 32 GiB.
- 1: Allows 1 GiB App Direct granularity which may result in App Direct region sizes that are not recommended.

#### **DBG\_LOG\_LEVEL**

Whether debug logging is enabled in the DCPMM host software. These logs pertain to the operation of the command-line tool only and do not reflect any logging functionality of the DCPMM. One of:

- 0: Logging is disabled. This is the default.
- 1: Log Errors.
- 2: Log Warnings, Errors.
- 3: Log Informational, Warnings, Errors.
- 4: Log Verbose, Informational, Warnings, Errors.

## **2.6.6. Change Preferences**

### **NAME**

ipmctl-change-preferences - Modifies one or more user preferences

### **SYNOPSIS**

```
ipmctl set [OPTIONS] -preferences [PROPERTIES]
```

### **DESCRIPTION**

Modifies one or more user preferences in the DCPMM software.

### **OPTIONS**

**-h**

**-help**

Displays help for the command.



## PROPERTIES

### CLI\_DEFAULT\_DIMM\_ID

The default display of DCPMM identifiers. One of:

- "UID": Use the DimmUID attribute as defined in the section **Show Device**.
- "HANDLE": Use the DimmHandle attribute as defined in section **Show Device**. This is the default

### CLI\_DEFAULT\_SIZE

The default display of capacities in the CLI. One of:

- "AUTO": Automatically choose the best format for each capacity in binary multiples of bytes (i.e., B, MiB, GiB or TiB). This is the default.
- "AUTO\_10": Automatically choose the best format for each capacity in decimal multiples of bytes (i.e., B, MB, GB or TB).
- "B": Displays all capacities in bytes.
- "MB": Displays all capacities in megabytes.
- "MiB": Displays all capacities in mebibytes.
- "GB": Displays all capacities in gigabytes.
- "GiB": Displays all capacities in gibibytes.
- "TB": Displays all capacities in terabytes.
- "TiB": Displays all capacities in tebibytes.

### APPDIRECT\_SETTINGS

The interleave settings to use when creating App Direct capacity in the format: (IMCSize ChannelSize). Must be one of the BIOS supported App Direct settings returned by the command **Show System Capabilities**.

NOTE	ByOne is not a valid setting for this preference. The default is "RECOMMENDED" which uses the BIOS recommended App Direct settings.
NOTE	The same interleave settings are used for all the App Direct capacity in the system. Therefore, if any App Direct capacity already exists, this preference cannot be changed.

### APPDIRECT\_GRANULARITY

The minimum App Direct granularity per DCPMM supported by the command



**Create Memory Allocation Goal.** One of:

- "RECOMMENDED": Use the recommended App Direct granularity of 32 GiB.
- "1": Allows 1 GiB App Direct granularity which may result in App Direct region sizes that are not recommended.

## EXAMPLES

Use DimmUID as the default DCPMM identifier, and display all capacities in bytes.

```
ipmctl set -preferences CLI_DEFAULT_DIMM_ID=UID  
CLI_DEFAULT_SIZE=B
```

## RETURN DATA

Returns the status of the operation.

## SAMPLE OUTPUT

```
Set (Property)=(Value): Success | Error (Code) -  
(Description)  
...
```

# 2.7. Debug

## 2.7.1. Run Diagnostic

### NAME

ipmctl-run-diagnostic - Runs a diagnostic test

### SYNOPSIS

```
ipmctl start [OPTIONS] -diagnostic  
(Quick|Config|Security|FW) -dimm(DIMMIDs)
```

### DESCRIPTION

Runs a diagnostic test.



## OPTIONS

**-h**

**-help**

Displays help for the command.

## TARGETS

**-diagnostic (Quick | Config | Security | FW)**

Run a specific test by supplying its name. By default all tests are run. One of:

- "Quick" - This test verifies that the DCPMM host mailbox is accessible and that basic health indicators can be read and are currently reporting acceptable values.
- "Config" - This test verifies that the BIOS platform configuration matches the installed hardware and the platform configuration conforms to best known practices.
- "Security" - This test verifies that all DCPMMs have a consistent security state. It's a best practice to enable security on all DCPMMs rather than just some.
- "FW" - This test verifies that all DCPMMs of a given model have consistent FW installed and other FW modifiable attributes are set in accordance with best practices.

Note that the test does not have a means of verifying that the installed FW is the optimal version for a given DCPMM model just that it's been consistently applied across the system.

**-dimm [(DimmIDS)]**

Runs a diagnostic test on specific DCPMMs by optionally supplying one or more comma-separated DCPMM identifiers. The default is to run the specified tests on all manageable DCPMMs. Only valid for the Quick diagnostic test.

## EXAMPLES

Runs all diagnostics.

```
ipmctl start -diagnostic
```

Runs the quick check diagnostic on DCPMM 0x0001

```
ipmctl start -diagnostic Quick -dimm 0x0001
```



## LIMITATIONS

If a DCPMM is unmanageable, then Quick test will report the reason, while Config, Security and FW tests will skip unmanageable DCPMMs.

## RETURN DATA

Each diagnostic generates one or more log messages. A successful test generates a single log message per DCPMM indicating that no errors were found. A failed test might generate multiple log messages each highlighting a specific error with all the relevant details. Each log contains the following information.

### TestName

The test name. One of:

- "Quick"
- "Config"
- "Security"
- "FW"

### State

The severity of the error. One of:

- "Ok"
- "Warning"
- "Failed"
- "Aborted"

NOTE: State is promoted to the highest severity result from the test group.

### Message

A free form textual description of the error.

## 2.7.2. Show Error Log

### NAME

`ipmctl-show-error-log` - Shows thermal or media errors on the specified DCPMMs





## SYNOPSIS

```
ipmctl show [OPTIONS] -error (Thermal|Media)
[TARGETS] [PROPERTIES]
```

## DESCRIPTION

Shows thermal or media errors on the specified DCPMMs.

## OPTIONS

**-h**

**-help**

Displays help for the command.

## TARGETS

**-dimm (DimmIDs)**

Filter output to events on a specific DCPMM by supplying the dimm target and one DCPMM identifier.

## PROPERTIES

### SequenceNumber

Error log entries are stored with a sequence number starting with 1 and rolling over back to 1 after 65535. Limit the error log entries returned by providing a sequence number. Only errors with a sequence number equal to or higher than provided will be returned. The default is 1.

### Level

Severity level of errors to be fetched. One of:

- "High": High severity errors (Default)
- "Low": Low severity errors

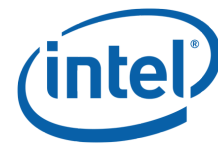
### Count

Max number of error entries to be fetched and printed. The default is 8 for media errors and 16 for thermal errors.

## EXAMPLES

Show all high thermal error log entries

```
ipmctl show -error Thermal Level=High
```



Show all low media error log entries

```
ipmctl show -error Thermal Level=High
```

## LIMITATIONS

To successfully execute this command, the specified DCPMM(s) must be manageable by the host software.

## RETURN DATA

Prints errors of the specified type for the specified DCPMMs. If no errors are found, the following message will be printed:

```
No errors found on DIMM (DimmID)
```

## SAMPLE OUTPUT

```
Thermal Error occurred on Dimm (DimmID):  
System Timestamp : 1527273299  
Temperature : 88C  
Reported : 4 - Critical  
Temperature Type : 0 - Media Temperature  
Sequence Number : 1
```

```
Media Error occurred on Dimm (DimmID):  
System Timestamp : 1527266471  
DPA : 0x000014c0  
PDA : 0x00000600  
Range : 4B  
Error Type : 4 - Locked/Illegal Access  
Error Flags : DPA Valid  
Transaction Type : 11 - CSR Write  
Sequence Number : 2
```

## 2.7.3. Dump Debug Log

### NAME

`ipmctl-dump-debug-log` - Dumps encoded firmware debug logs from specified DCPMMs and optionally decodes to human readable text.



## SYNOPSIS

```
ipmctl dump [OPTIONS] -destination (file_prefix) [-dict (file)] -debug -dimm (DimmIDs) [PROPERTIES]
```

## DESCRIPTION

Dumps encoded firmware debug logs from specified DCPMMs and optionally decodes to human readable text.

**NOTE** For any non-functional DCPMMs logs will be retrieved via SMBus.

## OPTIONS

**-h**

**-help**

Displays help for the command.

## TARGET

**-destination (file\_prefix)**

The command will create files that use the given filename as a prefix and append the DCPMM UID, DCPMM handle, debug log source, and the appropriate file type (.bin for encoded logs, .txt for decoded logs) onto the end.

```
file_prefix_Uid_Handle_logsource.[bin,txt]
```

**-dict (path)**

Optional file path to the dictionary file. If supplied, the command will create both the binary debug log and a text file with decoded log data with the file prefix specified by -destination.

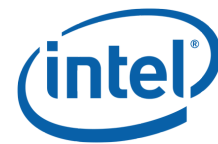
**-dimm (DimmIDs)**

Dumps the debug logs from the specified DCPMMs.

## EXAMPLES

Dumps and decodes the debug log from DCPMM 0x0001 and 0x0011 using the dictionary file.

```
ipmctl dump -destination file_prefix -dict  
nlog_dict.txt -debug -dimm 0x0001,0x0011
```



## LIMITATIONS

To successfully execute this command, the specified DCPMM(s) must be manageable by the host software.

## RETURN DATA

Dumps the encoded and optionally decoded contents of all 3 firmware debug log sources for the specified DCPMMs. Output file names are generated based on the -destination parameter above.

## SAMPLE OUTPUT

```
Dumped media FW debug logs to file
(file_prefix_8089-A1-1816-
00000016_0x0001_media.bin)
Decoded 456 records to file (file_prefix_8089-A1-
1816-00000016_0x0001_media.txt)
No spi FW debug logs found
```

# 2.7.4. Show ACPI Tables

## NAME

ipmctl-show-acpi - Shows the system ACPI tables related to the DCPMMs

## SYNOPSIS

```
ipmctl show [OPTIONS] -system (NFIT|PCAT|PMTT)
```

## DESCRIPTION

Shows the system ACPI tables related to the DCPMMs in the system.

## OPTIONS

**-h**

**-help**

Displays help for the command.

## TARGETS

**-system (NFIT|PCAT|PMTT)**



The system ACPI table(s) to display. By default both the NFIT and PCAT tables are displayed. One of:

- "NFIT" - The NVDIMM Firmware Interface Table
- "PCAT" - The Platform Capabilities Table
- "PMTT" - The Platform Memory Topology Table

Refer to the ACPI specification for detailed information about the ACPI tables.

## EXAMPLES

Show the ACPI NFIT

```
ipmctl show -system NFIT
```

## RETURN DATA

Returns the formatted data from the requested ACPI tables and their sub-tables. Refer to the ACPI specification for detailed information about the format of the ACPI tables.

**NOTE** All data is presented in ACPI little endian format.

# 2.7.5. Show Device Platform Configuration Data

## NAME

ipmctl-show-pcd - Shows the platform configuration data for one or more DCPMMs

## SYNOPSIS

```
ipmctl show [OPTIONS] -dimm (DimmIds) -pcd  
(Config|LSA)
```

## DESCRIPTION

Shows the platform configuration data for one or more DCPMMs.

## OPTIONS

**-h**

**-help**



Displays help for the command.

## TARGETS

### **-dimm (DimmIDs)**

Restricts output to the platform configuration data on specific DCPMMs by supplying one or more comma-separated DCPMM identifiers. The default is to display the platform configuration data for all manageable DCPMMs.

### **-pcd (Config|LSA)**

Restricts output to a specific partition of the platform configuration data. The default is to display both. One of:

- Config - Configuration management information
- LSA - Namespace label storage area

## EXAMPLES

Shows the configuration information from the platform configuration data for all manageable DCPMMs.

```
ipmctl show -dimm -pcd
```

Shows the configuration information from the platform configuration data for DCPMM 0x1.

```
ipmctl show -dimm -pcd Config
```

## LIMITATIONS

The specified DCPMM(s) must be manageable by the host software.

## RETURN DATA

Returns the formatted data from the requested platform configuration data for the specified DCPMMs for debugging and troubleshooting purposes.

## 2.7.6. Delete Device Platform Configuration Data



## NAME

ipmctl-delete-pcd - Clears select partition data from the PCD

## SYNOPSIS

```
ipmctl delete [OPTIONS] -dimm (DimmIds) -pcd  
(Config|LSA)
```

## DESCRIPTION

When LSA is specified, the namespace label storage area partition in the platform configuration data from one or more DCPMMs is cleared. This is a destructive operation which will clear the entire namespace label storage area including all namespaces labels and the namespace label index block in order to re-purpose the DCPMM(s) for use in a different operating system. All data on any deleted namespace(s) becomes inaccessible.

### NOTE

Deleting PCD LSA partition data removes any logical OS namespace mapping to the persistent memory data, but does not explicitly delete or modify user data found in persistent memory.

When Config is specified, the Current, Input, and Output Data Size and Start Offset values in the Configuration header are set to zero, making those tables invalid.

### NOTE

When Config is specified, only PCD partition 1 is modified. If the platform is rebooted prior to creating a new goal on any targeted DCPMMs, UEFI platform firmware will detect the missing tables and, if possible, restore previous config using the PCD partition 0 tables.

### NOTE

This action can be useful when moving DCPMMs from one system to another, as goal creation rules may restrict provisioning dimms with an existing configuration. Deleting the PCD can be used as a way to prepare a DCPMM for provisioning with the create -goal or load -goal commands by clearing existing configuration metadata. This allows the DCPMM to be provisioned in isolation using the create-goal command with the -dimm option. Once the PCD has been deleted a new goal should be created before rebooting.



## WARNING

**This command may result in data loss. Data should be backed up to other storage before executing this command. Because of data dependencies, other commands may be affected until the system has been rebooted.**

## OPTIONS

**-f**

**-force**

Deleting the PCD data is a destructive operation which requires confirmation from the user for each DCPMM. This option suppresses the confirmation.

**-h**

**-help**

Displays help for the command.

## TARGETS

**-dimm (DimmIDs)**

Deletes the PCD data on specific DCPMMs by supplying one or more comma-separated DCPMM identifiers. The default is to delete the PCD data for all manageable DCPMMs.

**-pcd (Config|LSA)**

Restricts clearing select partition data in the platform configuration data area. The default is to clear both. One of:

- Config - Configuration management information
- LSA - Namespace label storage area

## EXAMPLES

Clears the namespace label storage area from all manageable DCPMMs

```
delete -dimm -pcd LSA
```

Clears the Cin, Cout, Ccur tables from all manageable DCPMMs

```
delete -dimm -pcd Config
```

## LIMITATIONS

The specified DCPMM(s) must be manageable by the host software, and if data-at-rest





security is enabled, the DCPMMs must be unlocked. Any existing namespaces associated with the requested DCPMM(s) should be deleted before running this command.

## RETURN DATA

For each DCPMM, the CLI will indicate the status of the operation. If a failure occurs when deleting the platform configuration data from multiple DCPMMs, the process will continue deleting the remaining DCPMMs.

## 2.7.7. Show SMBIOS Tables

### NAME

`ipmctl-show-smbios` - Shows the system SMBIOS tables related to the DCPMMs.

### SYNOPSIS

```
ipmctl show [OPTIONS] -smbios (17|20) [TARGETS]
```

### DESCRIPTION

Shows the system SMBIOS tables related to the DCPMMs in the system.

### OPTIONS

**-h**

**-help**

Displays help for the command.

### TARGETS

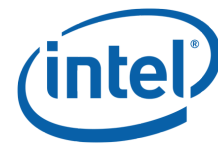
**-smbios (17|20)**

The SMBIOS table to display. Limit to a specific table by supplying the table type (from the SMBIOS specification). One of:

- 17 - Memory Device (Type 17)
- 20 - Memory Device Mapped Address (Type 20)

**-dimm (DimmIDs)**

Restricts output to specific DIMMs by supplying the DIMM target and one or more comma-separated DIMM identifiers. The default is to display all DIMMs.



## EXAMPLES

Show the type 17 SMBIOS table for all DCPMM

```
ipmctl show -smbios 17
```

## LIMITATIONS

Command available with debug driver only.

## RETURN DATA

Dumps the contents of the SMBIOS tables for each DCPMM. Refer to the SMBIOS specification Memory Device (Type 17) and Memory Device Mapped Address (Type 20) sections for more information.