
SSF for MaxCore™ MC3000 Platform

Command Line Interface Guide

P/N: 6806800T87H

March 2020



© 2020 SMART Embedded Computing™, Inc.

All Rights Reserved.

Trademarks

The stylized "S" and "SMART" is a registered trademark of SMART Modular Technologies, Inc. and "SMART Embedded Computing" and the SMART Embedded Computing logo are trademarks of SMART Modular Technologies, Inc. All other names and logos referred to are trade names, trademarks, or registered trademarks of their respective owners. These materials are provided by SMART Embedded Computing as a service to its customers and may be used for informational purposes only.

Disclaimer*

SMART Embedded Computing (SMART EC) assumes no responsibility for errors or omissions in these materials. **These materials are provided "AS IS" without warranty of any kind, either expressed or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.** SMART EC further does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SMART EC shall not be liable for any special, indirect, incidental, or consequential damages, including without limitation, lost revenues or lost profits, which may result from the use of these materials. SMART EC may make changes to these materials, or to the products described therein, at any time without notice. SMART EC makes no commitment to update the information contained within these materials.

Electronic versions of this material may be read online, downloaded for personal use, or referenced in another document as a URL to a SMART EC website. The text itself may not be published commercially in print or electronic form, edited, translated, or otherwise altered without the permission of SMART EC.

It is possible that this publication may contain reference to or information about SMART EC products, programming, or services that are not available in your country. Such references or information must not be construed to mean that SMART EC intends to announce such SMART EC products, programming, or services in your country.

Limited and Restricted Rights Legend

If the documentation contained herein is supplied, directly or indirectly, to the U.S. Government, the following notice shall apply unless otherwise agreed to in writing by SMART Embedded Computing.

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data clause at DFARS 252.227-7013 (Nov. 1995) and of the Rights in Noncommercial Computer Software and Documentation clause at DFARS 252.227-7014 (Jun. 1995).

SMART Embedded Computing, Inc.

2900 S. Diablo Way, Suite 190

Tempe, Arizona 85282

USA

*For full legal terms and conditions, visit www.smartembedded.com/ec/legal

Table of Contents

| | |
|---|-----------|
| About this Manual | 11 |
| 1 Introduction | 15 |
| 1.1 Accessing SSF using CLI | 15 |
| 1.2 Connecting to CLI | 16 |
| 2 Access Control Commands | 19 |
| 2.1 show | 19 |
| 2.2 user | 20 |
| 2.3 group | 21 |
| 2.4 show-all | 21 |
| 2.5 user, group, and membership | 22 |
| 3 Configuration Commands | 23 |
| 3.1 snapshot | 23 |
| 3.2 reload | 23 |
| 3.3 commit | 24 |
| 3.4 system-upgrade-initiate | 24 |
| 3.4.1 system-upgrade-status | 25 |
| 4 Hardware Platform Manager Commands | 29 |
| 4.1 applyPEXConfiguration | 31 |
| 4.2 assignIODeviceFunctionToCPU | 31 |
| 4.3 createIODeviceFunctions | 32 |
| 4.4 destroyAllIODeviceFunctions | 33 |
| 4.5 eventLog | 33 |
| 4.6 getCPUAssignedFunctions | 34 |
| 4.7 getCPUVirtualNICFunctionCount | 34 |
| 4.8 getChassisPayloadPowerPolicy | 35 |
| 4.9 getCoolingPolicy | 35 |
| 4.10 getFunctionAssignedCPUAddress | 35 |
| 4.11 getPEXMode | 36 |
| 4.12 getSATAAssignment | 36 |
| 4.13 getShelfHostAddress | 37 |

Table of Contents

| | | |
|------|-------------------------------------|----|
| 4.14 | getUSBAssignment | 37 |
| 4.15 | getZoneType | 37 |
| 4.16 | listAlarms | 38 |
| 4.17 | listAvailableCPUs | 38 |
| 4.18 | listAvailableIODevices | 38 |
| 4.19 | listShelfInfo | 39 |
| 4.20 | removeIODeviceFunctionFromCPU | 39 |
| 4.21 | resetPEXConfiguration | 40 |
| 4.22 | setCPUVirtualNICFunctionCount | 40 |
| 4.23 | setChassisPayloadPowerPolicy | 41 |
| 4.24 | setCoolingPolicy | 41 |
| 4.25 | setPEXMode | 42 |
| 4.26 | setSATAAssignment | 42 |
| 4.27 | setUSBAssignment | 43 |
| 4.28 | getBIOSBootOrder | 43 |
| 4.29 | getBIOSConsoleRedirectionParameters | 44 |
| 4.30 | getBIOSBootConfiguration | 44 |
| 4.31 | getBIOSPeripheralConfiguration | 45 |
| 4.32 | getBIOSIIOConfiguration | 45 |
| 4.33 | getBIOSSATAConfiguration | 46 |
| 4.34 | getBIOSVideoConfiguration | 46 |
| 4.35 | getBIOSUSBConfiguration | 47 |
| 4.36 | getBIOSProcessorConfiguration | 47 |
| 4.37 | getBIOSPowerManagementConfiguration | 48 |
| 4.38 | getBIOSThermalConfiguration | 48 |
| 4.39 | getBIOSMemoryConfiguration | 49 |
| 4.40 | getBIOSCommonRefcodeConfiguration | 49 |
| 4.41 | setBIOSBootOrderNew | 50 |
| 4.42 | setBIOSBootOrderDefault | 50 |
| 4.43 | setBIOSBootOrderbyIndex | 51 |
| 4.44 | setBIOSConsoleRedirectionParameters | 52 |
| 4.45 | setBIOSBootConfiguration | 52 |
| 4.46 | setBIOSPeripheralConfiguration | 53 |
| 4.47 | setBIOSIIOConfiguration | 54 |
| 4.48 | setBIOSSATAConfiguration | 55 |
| 4.49 | setBIOSVideoConfiguration | 55 |
| 4.50 | setBIOSUSBConfiguration | 56 |
| 4.51 | setBIOSProcessorConfiguration | 57 |

| | | |
|----------|--|-----------|
| 4.52 | setBIOSPowerManagementConfiguration | 58 |
| 4.53 | setBIOSThermalConfiguration | 59 |
| 4.54 | setBIOSMemoryConfiguration | 59 |
| 4.55 | setBIOSCommonRefcodeConfiguration | 60 |
| 4.56 | powerOffAllApplicationCPUs | 60 |
| 4.57 | setShelfAddress | 61 |
| 4.58 | getShelfAddress | 62 |
| 4.59 | setBMCIPAddress | 62 |
| 4.60 | getBMCIPAddress | 63 |
| 4.61 | listShelves | 63 |
| 4.62 | addShelf | 63 |
| 4.63 | removeShelf | 64 |
| 4.64 | modifyShelf | 64 |
| 4.65 | setSsfHostInterfaceName | 65 |
| 4.66 | getSsfHostInterfaceName | 65 |
| 4.67 | setDefaultConfiguration | 66 |
| 4.68 | getDiscoveryStatus | 66 |
| 4.69 | setPXEBootEnvironmentForCPU | 66 |
| 4.70 | getPXEBootConfigurationForCPU | 68 |
| 5 | System Commands | 69 |
| 5.1 | CLI Commands in disable mode | 69 |
| 5.2 | CLI commands in enable mode | 70 |
| 5.3 | CLI commands in config mode | 72 |
| 5.4 | CLI Commands in System Mode | 75 |
| 5.5 | exit | 76 |
| 5.6 | CLI Commands for Shelf Mode | 76 |
| 5.7 | CLI Commands for PCIeSlot Mode | 78 |
| 5.8 | CLI Commands for PCIeCard Mode | 79 |
| 5.9 | CLI Commands for CPU Mode | 80 |
| 5.10 | CLI Commands for BIOS Configuration mode | 81 |
| 5.11 | CLI Commands for VEE mode | 85 |
| 5.12 | CLI Commands for Linux application mode | 87 |
| 6 | System Log Collection Commands | 89 |
| 6.1 | collectlog | 89 |
| 6.2 | listlog | 89 |

Table of Contents

| | | |
|----------|--|-----------|
| 6.3 | deletelog | 90 |
| 6.4 | ssflogsize | 90 |
| 6.5 | downloadlog | 90 |
| A | Related Documentation | 91 |
| A.1 | SMART Embedded Computing Documentation | 91 |

List of Tables

| | | |
|------------|--|----|
| Table 1-1 | Command Line Editing Features | 16 |
| Table 2-1 | Parameters of the show command | 20 |
| Table 2-2 | Parameters of the user command | 20 |
| Table 2-3 | Parameters of the group command | 21 |
| Table 2-4 | Parameters of the user, group, and membership commands | 22 |
| Table 3-1 | Parameters of snapshot command | 23 |
| Table 3-2 | Parameters of reload command | 23 |
| Table 3-3 | Parameters of commit command | 24 |
| Table 3-4 | Parameters of systemupgrade command | 24 |
| Table 4-1 | Hardware Platform Manager Commends List | 29 |
| Table 4-2 | Parameters of applyPEXConfiguration | 31 |
| Table 4-3 | Parameters of assignIODeviceFunctionToCPU | 32 |
| Table 4-4 | Parameters of createIODeviceFunctions | 32 |
| Table 4-5 | Parameters of destroyAllIODeviceFunctions | 33 |
| Table 4-6 | Parameters of eventLog | 33 |
| Table 4-7 | Parameters of getCPUAssignedFunctions | 34 |
| Table 4-8 | Parameters of getCPUVirtualNICFunctionCount | 34 |
| Table 4-9 | Parameters of getChassisPayloadPowerPolicy | 35 |
| Table 4-10 | Parameters of getCoolingPolicy | 35 |
| Table 4-11 | Parameters of getFunctionAssignedCPUAddress | 35 |
| Table 4-12 | Parameters of getPEXMode | 36 |
| Table 4-13 | Parameters of getSATAAssignment | 36 |
| Table 4-14 | Parameters of getShelfHostAddress | 37 |
| Table 4-15 | Parameters of getUSBAssignment | 37 |
| Table 4-16 | Parameters of getZoneType | 37 |
| Table 4-17 | Parameters of listAlarms | 38 |
| Table 4-18 | Parameters of listAvailableCPUs | 38 |
| Table 4-19 | Parameters of listAvailableIODevices | 38 |
| Table 4-20 | Parameters of listShelfInfo | 39 |
| Table 4-21 | Parameters of removeIODeviceFunctionFromCPU | 39 |
| Table 4-22 | Parameters of resetPEXConfiguration | 40 |
| Table 4-23 | Parameters of setCPUVirtualNICFunctionCount | 40 |
| Table 4-24 | Parameters of removeIODeviceFunctionFromCPU | 41 |
| Table 4-25 | Parameters of setCoolingPolicy | 41 |
| Table 4-26 | Parameters of setPEXMode | 42 |
| Table 4-27 | Parameters of setSATAAssignment | 42 |
| Table 4-28 | Parameters of setUSBAssignment | 43 |

List of Tables

| | | |
|------------|---|----|
| Table 4-29 | Parameters of getBIOSBootOrder | 43 |
| Table 4-30 | Parameters of getBIOSConsoleRedirectionParameters | 44 |
| Table 4-31 | Parameters of getBIOSBootConfiguration | 44 |
| Table 4-32 | Parameters of getBIOSPeripheralConfiguration | 45 |
| Table 4-33 | Parameters of getBIOSIIOConfiguration | 45 |
| Table 4-34 | Parameters of getBIOS SATAConfiguration | 46 |
| Table 4-35 | Parameters of getBIOSVideoConfiguration | 46 |
| Table 4-36 | Parameters of getBIOSUSBConfiguration | 47 |
| Table 4-37 | Parameters of getBIOSProcessorConfiguration | 47 |
| Table 4-38 | Parameters of getBIOSPowerManagementConfiguration | 48 |
| Table 4-39 | Parameters of getBIOSThermalConfiguration | 48 |
| Table 4-40 | Parameters of getBIOSMemoryConfiguration | 49 |
| Table 4-41 | Parameters of getBIOSCommonRefcodeConfiguration | 49 |
| Table 4-42 | Parameters of setBIOSBootOrderNew | 50 |
| Table 4-43 | Parameters of setBIOSBootOrderDefault | 51 |
| Table 4-44 | Parameters of setBIOSBootOrderbyIndex | 51 |
| Table 4-45 | Parameters of setBIOSConsoleRedirectionParameters | 52 |
| Table 4-46 | Parameters of setBIOSBootConfiguration | 52 |
| Table 4-47 | Parameters of setBIOSPeripheralConfiguration | 53 |
| Table 4-48 | Parameters of setBIOSIIOConfiguration | 54 |
| Table 4-49 | Parameters of setBIOS SATAConfiguration | 55 |
| Table 4-50 | Parameters of getBIOSVideoConfiguration | 56 |
| Table 4-51 | Parameters of setBIOSUSBConfiguration | 56 |
| Table 4-52 | Parameters of setBIOSProcessorConfiguration | 57 |
| Table 4-53 | Parameters of setBIOSPowerManagementConfiguration | 58 |
| Table 4-54 | Parameters of setBIOSThermalConfiguration | 59 |
| Table 4-55 | Parameters of setBIOSMemoryConfiguration | 59 |
| Table 4-56 | Parameters of setBIOSCommonRefcodeConfiguration | 60 |
| Table 4-57 | Parameters of powerOffAllApplicationCPUs | 60 |
| Table 4-58 | Parameters of setShelfAddress | 61 |
| Table 4-59 | Parameters of getShelfAddress | 62 |
| Table 4-60 | Parameters of setBMCIPAddress | 62 |
| Table 4-61 | Parameters of getBMCIPAddress | 63 |
| Table 4-62 | Parameters of addShelf | 63 |
| Table 4-63 | Parameters of removeShelf | 64 |
| Table 4-64 | Parameters of modifyShelf | 64 |
| Table 4-65 | Parameters of setSsfHostInterfaceName | 65 |
| Table 4-66 | Parameters of setDefaultConfiguration | 66 |

List of Tables

| | | |
|------------|---|----|
| Table 4-67 | Parameters of getDiscoveryStatus | 66 |
| Table 4-68 | Parameters of getDiscoveryStatus | 67 |
| Table 4-69 | Parameters of getDiscoveryStatus | 68 |
| Table 5-1 | Parameters of Disable mode commands | 70 |
| Table 5-2 | Parameters of Enable mode commands | 71 |
| Table 5-3 | Parameters of Configure mode commands | 74 |
| Table 5-4 | Parameters of System mode commands | 75 |
| Table 5-5 | Parameters of Shelf mode commands | 76 |
| Table 5-6 | Parameters of PCIeSlot mode commands | 79 |
| Table 5-7 | Parameters of PCIeCard mode commands | 79 |
| Table 5-8 | Parameters of CPU mode commands | 80 |
| Table 5-9 | Parameters of BIOS mode commands | 82 |
| Table 5-10 | Parameters of VEE mode commands | 86 |
| Table 5-11 | Parameters of Linux Application mode commands | 88 |
| Table A-1 | SMART Embedded Computing Publications | 91 |

List of Tables

About this Manual

Overview of Contents

This guide provides detailed information on usage of Command Line Interface (CLI) to work with System Services Framework (SSF). The following list gives an overview of the chapters described in this document.

Chapter 1, Introduction provides overview of SSF, SSF key features, accessing of SSF using CLI, and connecting to CLI.

Chapter 2, Access Control Commands provides Access Control commands and their definitions in detail.

Chapter 3, Configuration Commands provides Snapshot, Reload, and Commit commands and their definitions in detail.

Chapter 4, Hardware Platform Manager Commands provides Hardware Platform Manager commands and their definitions in detail.

Chapter 5, System Commands provides System related commands and their definitions in detail.

Chapter 6, System Log Collection Commands provides System Log Collection commands and their definitions in detail. Also, describes how to collect system logs from different cards to SSF Core.

Appendix A, Related Documentation lists the relevant manuals.




Abbreviations





The following table lists the abbreviations used in this guide.

| Abbreviation | Definition |
|--------------|---|
| BMC | Baseboard Management Controller |
| CSIM | Common System Information Model |
| HPM | Hardware Platform Manager |
| PEE | Physical Execution Environment |
| PCIe | Peripheral Component Interconnect Express |
| VEE | Virtual Execution Environment |
| SRstackware | Switching and Routing stackware |
| SSF | System Services Framework |
| VEE | Virtual Execution Environment |
| VF | Virtual Function |

Conventions

The following table describes the conventions used throughout this manual.

| Notation | Description |
|---|---|
| 0x00000000 | Typical notation for hexadecimal numbers (digits are 0 through F), for example used for addresses and offsets |
| 0b0000 | Same for binary numbers (digits are 0 and 1) |
| bold | Used to emphasize a word |
| Screen | Used for on-screen output and code related elements or commands. Sample of Programming used in a table (9pt) |
| Courier + Bold | Used to characterize user input and to separate it from system output |
| <i>Reference</i> | Used for references and for table and figure descriptions |
| File > Exit | Notation for selecting a submenu |
| <text> | Notation for variables and keys |
| [text] | Notation for software buttons to click on the screen and parameter description |
| ... | Repeated item for example node 1, node 2, ..., node 12 |
| . | Omission of information from example/command that is not necessary at the time |
| .. | Ranges, for example: 0..4 means one of the integers 0,1,2,3, and 4 (used in registers) |
| | Logical OR |
|  | Indicates a hazardous situation which, if not avoided, could result in death or serious injury |
|  | Indicates a hazardous situation which, if not avoided, may result in minor or moderate injury |
|  | Indicates a property damage message |

| Notation | Description |
|---|--|
|  | Indicates a hot surface that could result in moderate or serious injury |
|  | Indicates an electrical situation that could result in moderate injury or death |
| Use ESD protection  | Indicates that when working in an ESD environment care should be taken to use proper ESD practices |
|  | No danger encountered, pay attention to important information |

Summary of Changes

| Part Number | Date | Description |
|-------------|--------------|---|
| 6806800T87H | March 2020 | Rebranded to SMART Embedded Computing. |
| 6806800T87G | January 2018 | Added a new command, power policy in Table 5-8 . |
| 6806800T87F | July 2017 | Added sections setDefaultConfiguration , getDiscoveryStatus , setPXEBootEnvironmentForCPU , and getPXEBootConfigurationForCPU . |
| 6806800T87E | April 2017 | Updated CLI commands in config mode on page 72 . Added sections powerOffAllApplicationCPUs , setShelfAddress , getShelfAddress , setBMCIPAddress , getBMCIPAddress , listShelves , addShelf , removeShelf , modifyShelf , setSsfHostInterfaceName . |
| 6806800T87D | March 2017 | Added CLI Commands for PCIeSlot Mode , CLI Commands for PCIeCard Mode , CLI Commands for CPU Mode , CLI Commands for BIOS Configuration mode , CLI Commands for VEE mode on page 85 , CLI Commands for Linux application mode , setBIOSBootOrderDefault , setBIOSBootOrderbyIndex , and system-upgrade-initiate . |

| Part Number | Date | Description |
|--------------------|----------------|---|
| 6806800T87C | November 2016 | Updated Chapter 1, Introduction , Chapter 2, Access Control Commands , Chapter 3, Configuration Commands , Chapter 4, Hardware Platform Manager Commands , and Chapter 5, System Commands . |
| 6806800T87B | September 2016 | Added a new chapter SharpMedia PCIE-8120 Commands . |
| 6806800T87A | January 2016 | Initial version |

Introduction

System Services Framework (SSF) is a management and configuration interface to Artesyn's hardware and software products. It facilitates system level configuration and management access to SSF managed hardware and software components, through Web, CLI, and XML protocol interfaces.

SSF learns all the hardware and system software components to represent in Common System Information Model (CSIM). It provides a user interface to manage servers from a single management interface and stores the attributes and corresponding configurations of each hardware or software component in the PostgreSQL database for persistency.

SSF key features:

- Access, Authentication, and Authorization
- Configuration persistency, reload, and rollback
- Hierarchal representation of System Model
- Dynamic population of System Model
- Remote system configuration management, firmware upgrade, and system diagnosis

1.1 Accessing SSF using CLI

You can access SSF using the CLI. SSF provides a fully functional CLI with auto complete, history, and help features.

SSF CLI is a Telnet daemon that waits for inbound TCP connections. You can access the CLI through Telnet session. The CLI can serve multiple client sessions simultaneously, and the number of sessions supported by SSF is limited to `maxSessions` configured in `/opt/ssf/etc/config/main/cli.cfg`.

SSF CLI enables to traverse the system hierarchically and provides configuration and management access to the SSF system. All these hierarchies are represented as nodes.

Using CLI, you can:

- Edit SSF configuration
- View general system information
- Configure and manage all platform services, such as `syslog-ng` and `syslcu`
- Upgrade the firmware

Connecting to CLI

The following table provides command line editing features of SSF CLI.

Table 1-1 Command Line Editing Features

| Keywords | Description |
|---------------------------|---|
| Left and Right arrow keys | Allow you to move the cursor within the current command line. |
| Up and Down arrow keys | Allow you to browse through a command history. |
| BACKSPACE Key | Enables you to remove the character towards left. |
| TAB key | Completes the keyword being entered automatically. |
| "?" key | Provides you context help. |
| <cr> | Carriage return. System displays this command when all mandatory arguments of a particular CLI command are provided. It represents the command syntax completion. |

1.2 Connecting to CLI

You can connect to the CLI after login to the SSF running Host, using External SSH Daemon with SSH Connection. This is the default behavior of SSF CLI.

Start the **Telnet** connection from an already established secure shell.

```
[root@abts-ap-static-214 ~]# telnet localhost 11001
Trying ::1...
telnet: connect to address ::1: Connection refused
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Welcome to SSF CLI
Username: Admin
Password:
Access granted
>enable
#configure terminal
MaxCore(config)
```




By default, the IP address is configured as 127.0.0.1 and the port numbers are 11001(CLI) and 15550 (XML).

Also the cli.cfg file configures:

The maximum number of connections to the command line interface, default value is 5.

The time out in seconds, default value is 300.

Access Control Commands

The access control commands allow you to access the authentication mode of the SSF. Using the access control commands, you can:

- Add new users
- Modify existing user details
- Delete existing users
- Create new groups and add users into it
- View the list of available users and groups.

To execute the access control commands, you should work in the privilege mode of CLI as shown below:

```
[root@abts-ap-static-214 ~]# telnet localhost 11001
```

```
Trying ::1...
```

```
telnet: connect to address ::1: Connection refused
```

```
Trying 127.0.0.1...
```

```
Connected to localhost.
```

```
Escape character is '^]'.
```

```
Welcome to SSF CLI
```

```
Username: Admin
```

```
Password:
```

```
Access granted
```

```
>enable
```

```
#configure terminal
```

```
MaxCore(config)#
```

2.1 show

The `show` command in the authentication mode allows you to view the properties of users and groups. This command takes user name or group name as its parameters. When the parameter is `<user name>`, it displays the properties of the specified user name. When the parameter is `<group name>`, it displays the properties of the specified group name.

Syntax

```
#show {user|group} <username>|<groupname>
```

Table 2-1 Parameters of the show command

| Argument | Description |
|-------------|---|
| <username> | Type the valid user name of whom you want to view the details. The name and password of the specified user are displayed. |
| <groupname> | Type the valid group name of which you want to view the details. The name and membership object of the specified group are displayed. |

Example

```
#show user Admin
```

2.2 user

The user commands allow you to add, delete, and modify users from the user database.

To add a user:

Syntax

```
#username <username> password <password>
```

To modify a user password:

Syntax

```
#password <username> <password>
```

To delete a user:

Syntax

```
#no username <username>
```

Table 2-2 Parameters of the user command

| Argument | Description |
|------------|--|
| <username> | Type the valid user name to be added, modified, or deleted |
| <password> | Type the corresponding password of the specified user name |

2.3 group

The `group` command allows you to add new groups and delete existing groups.

To add a group:

Syntax

```
#group <groupname>
```

To delete a group:

Syntax

```
#no group <groupname>
```

Table 2-3 Parameters of the group command

| Argument | Description |
|-------------|--|
| <groupname> | Type the valid group name to be added or deleted |

2.4 show-all

The `show-all` command in authentication mode allows you to view the list of available users and groups. This command takes `user` or `group` as its parameters. When the parameter is `user`, it displays the list of all users added in SSF. When the parameter is `group`, it displays the list of all groups added in SSF.

Syntax

```
#show {user|group}
```

Example

```
#show user
```

2.5 user, group, and membership

The `user`, `group`, and `membership` commands allows you to add or delete users from the existing groups.

To add a user to a group

Syntax

```
#assign <username> to <groupname>
```

To delete a user from a group

Syntax

```
#remove <username> from <groupname>
```

Table 2-4 Parameters of the user, group, and membership commands

| Argument | Description |
|-------------|--|
| <username> | Type the valid user name to be added or deleted from a group |
| <groupname> | Type the valid group name to which a user to be added or deleted |

Configuration Commands

This chapter describes the configuration commands available with SSF.

3.1 snapshot

The snapshot command stores the complete configuration of a component and all the objects under its hierarchy, to a specified file. With disable option, it will not copy read only configuration to snapshot and with enable option, it will copy read only configuration to snapshot.

Syntax

```
#snapshot output <configuration file to store> <disable|enable>
```

Table 3-1 Parameters of snapshot command

| Argument | Description |
|-------------------------------|---|
| <configuration file to store> | Type the file name into which the configuration of component is stored. |
| <disable enable> | disable will not copy read only configuration to snapshot. enable will copy read only configuration to snapshot. |

3.2 reload

The reload command uploads the configuration of components collected at a particular hierarchy level, to the current configuration. This command is available at configuration level and any file that is stored using snapshot command can be reloaded. You can also modify the snapshot file before performing a reload operation.

Syntax

```
#reload script <configuration file to reload>
```

Table 3-2 Parameters of reload command

| Argument | Description |
|--------------------------------|--|
| <configuration file to reload> | Type the file name from which the configuration of component is to be reloaded to the current configuration. |

This command allows only to reload the snapshot taken/captured via CLI interface. In case of modifying the snapshot file before reload, make sure that inputs modified in the snapshot file are valid and syntactically correct. Any syntax errors or incorrect inputs may not result in any intimations or rollback.

3.3 commit

The `commit` command applies the changes made to the configuration file in the current configuration. This command supports rollback feature. In case of any failure while committing the configuration, system will be rolled back to the previous state.

Syntax

```
#commit configuration <Configuration Name>
```

Table 3-3 Parameters of commit command

| Argument | Description |
|----------------------|--|
| <Configuration Name> | Type the configuration name that you want to commit. |

3.4 system-upgrade-initiate

The `system-upgrade-initiate` CLI command allows user to upgrade at shelf level. This `system-upgrade-initiate` command avoids intervention during the software or firmware upgrades. The command will take single package as input to upgrade whole MaxCore system. The input package will be extracted and respective software or firmware will be sent to all upgradable devices or entities. Then the upgrade process will start in sequence to all the devices and entities in the hierarchy.

Syntax

```
#system-upgrade-initiate filename "<Absolute path>"
```

Table 3-4 Parameters of systemupgrade command

| Argument | Description |
|---------------|--------------------------------|
| Absolute path | Upgrade Package Absolute Path. |

Example

```
MaxCore(shelf-1-1)#system-upgrade-initiate filename "/root/MC_SYSTEMUPDATE-R_1.1.0.25-2.iso"
```

Sample Output

```
systemupgrade Initialized
```

```
Update initiated successfully. System may go for multiple reboots and this takes long time.
```

```
If connected, please use status command to get the current state of the update
```


3.4.1 system-upgrade-status

The system-upgrade-status CLI command allows user to check the status of ongoing upgrade at shelf level.

Syntax

```
#system-upgrade-status
```

Example

```
MaxCore(shelf-1-1)#system-upgrade-status
```

Sample Output

```
MaxCore(shelf-1-1)#system-upgrade-status
```

```
=====
```

```
Shelf# 1                status: Upgrade In Progress
```

```
=====
```

```
Slot# 1
```

```
CPU# 1  status: Upgrade In Progress
```

```
-----
```

| Entity | Status | Curr Ver | Last Updated |
|--------|--------|----------|--------------|
|--------|--------|----------|--------------|

```
-----
```

| | | | |
|-------------------------|--------------|------------|------------------------------|
| BMC_Application Upgrade | Not Required | 1.8.000000 | Wed Mar 08 06:16:22 IST 2017 |
|-------------------------|--------------|------------|------------------------------|

| | | | |
|----------------------------|--------------|---------------|------------------------------|
| BMC_CPLD_Companion Upgrade | Not Required | 0.03.03000000 | Wed Mar 08 06:16:22 IST 2017 |
|----------------------------|--------------|---------------|------------------------------|

| | | | |
|-----------------------|--------------|---------------|------------------------------|
| BMC_CPLD_Glue Upgrade | Not Required | 0.03.00000000 | Wed Mar 08 06:16:22 IST 2017 |
|-----------------------|--------------|---------------|------------------------------|

| | | | |
|-----------------------------|--------------|---------------|------------------------------|
| BMC_CPLD_I/O_Module Upgrade | Not Required | 0.06.01000000 | Wed Mar 08 06:16:22 IST 2017 |
|-----------------------------|--------------|---------------|------------------------------|

| | | | |
|------------------------------|--------------|-------|------------------------------|
| Basic_Board_Services Upgrade | Not Required | 0.3.3 | Wed Mar 08 06:16:22 IST 2017 |
|------------------------------|--------------|-------|------------------------------|

| | | | |
|-----------------------------------|-------------|----------|------------------------------|
| System_Services_Framework Upgrade | In Progress | 1.1.0.28 | Wed Mar 08 06:16:27 IST 2017 |
|-----------------------------------|-------------|----------|------------------------------|

| | | | |
|------------------------|--------------|--------------|------------------------------|
| PCIe-7210_BIOS Upgrade | Not Required | 2.1.00000005 | Wed Mar 08 06:16:22 IST 2017 |
|------------------------|--------------|--------------|------------------------------|

| | | | |
|------------------------|--------------|----------|------------------------------|
| PCIe-7210_CPLD Upgrade | Not Required | 01.05.03 | Wed Mar 08 06:16:22 IST 2017 |
|------------------------|--------------|----------|------------------------------|

| | | | |
|-------------------|---------|----------|------------------------------|
| ViewCheck Upgrade | Pending | 1.1.0.14 | Wed Mar 08 06:16:22 IST 2017 |
|-------------------|---------|----------|------------------------------|

system-upgrade-status

CPU# 2 status:--

Entity Status Curr Ver Last Updated

Slot# 2

Slot# 3

Slot# 4

Slot# 5

Slot# 6

Slot# 7

Slot# 8

Slot# 9

Slot# 10

Slot# 11

Slot# 12

Slot# 13

Slot# 14

Slot# 15

MaxCore(shelf-1-1)#system-upgrade-status

=====
Shelf# 1 status: Upgrade Success
=====

Slot# 1

CPU# 1 status: Upgrade Success

Entity Status Curr Ver Last Updated

BMC_Application -- 1.8.000000 Wed Mar 08 06:12:40 IST 2017

system-upgrade-status

```
BMC_CPLD_Companion  --  0.03.03000000    Wed Mar 08 06:12:40 IST 2017
BMC_CPLD_Glue      --  0.03.00000000    Wed Mar 08 06:12:40 IST 2017
BMC_CPLD_I/O_Module  --  0.06.01000000    Wed Mar 08 06:12:40 IST 2017
Basic_Board_Services  --  0.3.3            Wed Mar 08 06:12:40 IST 2017
System_Services_Framework  Upgrade Success 1.1.0.29 Wed Mar 08 06:18:41 IST
2017
PCIe-7210_BIOS      --  2.1.00000005    Wed Mar 08 06:12:40 IST 2017
PCIe-7210_CPLD      --  01.05.03        Wed Mar 08 06:12:40 IST 2017
ViewCheck           Upgrade Success  1.1.0.15    Wed Mar 08 06:18:55 IST 2017
CPU# 2  status:      --
```

```
-----
Entity      Status          Curr Ver      Last Updated
-----
```

Slot# 2

Slot# 3

Slot# 4

Slot# 5

Slot# 6

Slot# 7

Slot# 8

Slot# 9

Slot# 10

Slot# 11

Slot# 12

Slot# 13

Slot# 14

Slot# 15

MaxCore(shelf-1-1)#

system-upgrade-status

Hardware Platform Manager Commands

Hardware Platform Manager (HPM) is used to view and configure the MaxCore™ MC3000 platform. It allows you to choose shelf host, associate disks and USBs among all the CPUs available in slot 1 and slot 15, and lets you set and get the values of BIOS boot parameters of a CPU present in any slot (1 to 15) in the system. Following are the list of HPM commands.

Table 4-1 Hardware Platform Manager Commands List

| Command | Command |
|--------------------------------------|--|
| <i>applyPEXConfiguration</i> | <i>getBIOSProcessorConfiguration</i> |
| <i>assignIODeviceFunctionToCPU</i> | <i>getBIOSPowerManagementConfiguration</i> |
| <i>createIODeviceFunctions</i> | <i>getBIOSThermalConfiguration</i> |
| <i>destroyAllIODeviceFunctions</i> | <i>getBIOSMemoryConfiguration</i> |
| <i>eventLog</i> | <i>getBIOSCommonRefcodeConfiguration</i> |
| <i>getCPUAssignedFunctions</i> | <i>setBIOSBootOrderNew</i> |
| <i>getCPUVirtualNICFunctionCount</i> | <i>setBIOSBootOrderDefault</i> |
| <i>getChassisPayloadPowerPolicy</i> | <i>setBIOSBootOrderbyIndex</i> |
| <i>getCoolingPolicy</i> | <i>setBIOSConsoleRedirectionParameters</i> |
| <i>getFunctionAssignedCPUAddress</i> | <i>setBIOSBootConfiguration</i> |
| <i>getPEXMode</i> | <i>setBIOSPeripheralConfiguration</i> |
| <i>getSATAAssignment</i> | <i>setBIOSIIOConfiguration</i> |
| <i>getShelfHostAddress</i> | <i>setBIOSSATAConfiguration</i> |
| <i>getUSBAssignment</i> | <i>setBIOSVideoConfiguration</i> |
| <i>getZoneType</i> | <i>setBIOSUSBConfiguration</i> |
| <i>listAlarms</i> | <i>setBIOSProcessorConfiguration</i> |
| <i>listAvailableCPUs</i> | <i>setBIOSPowerManagementConfiguration</i> |
| <i>listAvailableIODevicees</i> | <i>setBIOSThermalConfiguration</i> |
| <i>listShelfInfo</i> | <i>setBIOSMemoryConfiguration</i> |
| <i>removeIODeviceFunctionFromCPU</i> | <i>setBIOSCommonRefcodeConfiguration</i> |

Table 4-1 Hardware Platform Manager Commands List (continued)

| Command | Command |
|--|--------------------------------------|
| <i>resetPEXConfiguration</i> | <i>powerOffAllApplicationCPUs</i> |
| <i>setCPUVirtualNICFunctionCount</i> | <i>setShelfAddress</i> |
| <i>setChassisPayloadPowerPolicy</i> | <i>getShelfAddress</i> |
| <i>setCoolingPolicy</i> | <i>setBMCIPAddress</i> |
| <i>setPEXMode</i> | <i>getBMCIPAddress</i> |
| <i>setSATAAssignment</i> | <i>listShelves</i> |
| <i>setUSBAssignment</i> | <i>addShelf</i> |
| <i>getBIOSBootOrder</i> | <i>removeShelf</i> |
| <i>getBIOSConsoleRedirectionParameters</i> | <i>modifyShelf</i> |
| <i>getBIOSBootConfiguration</i> | <i>setSsfHostInterfaceName</i> |
| <i>getBIOSPeripheralConfiguration</i> | <i>getSsfHostInterfaceName</i> |
| <i>getBIOSIIOConfiguration</i> | <i>setDefaultConfiguration</i> |
| <i>getBIOSSATAConfiguration</i> | <i>getDiscoveryStatus</i> |
| <i>getBIOSVideoConfiguration</i> | <i>setPXEBootEnvironmentForCPU</i> |
| <i>getBIOSUSBConfiguration</i> | <i>getPXEBootConfigurationForCPU</i> |

NOTICE

All the commands for changing BIOS boot parameters require a power cycle of the system for the changes to take effect.

To execute the HPM commands, you should enter into the privilege mode of CLI as shown below:

```
[root@abts-ap-static-214 ~]# telnet localhost 11001
```

```
Trying ::1...
```

```
telnet: connect to address ::1: Connection refused
```

```
Trying 127.0.0.1...
```

```
Connected to localhost.
```

```
Escape character is '^]'.
Welcome to SSF CLI
Username: Admin
Password:
Access granted
>enable
#configure terminal
MaxCore(config)# HardwarePlatformManager
```

4.1 applyPEXConfiguration

Applies the PEX configuration provided by user. It restarts PEX and power cycles PCIe slots. Then it runs the configuration software on the Baseboard Management Controller (BMC).

NOTICE

Before applying configuration, shutdown all application CPUs because system will go for power cycle. For more details, refer section [powerOffAllApplicationCPUs](#).

Syntax

```
#applyPEXConfiguration rackID <Rack number> ShelfID <Shelf number>
```

Table 4-2 Parameters of applyPEXConfiguration

| Argument | Description |
|--------------|------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis |

4.2 assignIODeviceFunctionToCPU

Assigns the function of an I/O device to the PCI tree of a CPU.

Syntax

```
#assignIODeviceFunctionToCPU cpuAddress <CPU address> rackID <Rack number> ShelfID <Shelf number> vfLocation <Virtual Function Location>
```

createIODeviceFunctions

Table 4-3 Parameters of assignIODeviceFunctionToCPU

| Argument | Description |
|---------------------------|--|
| CPU address | Location of CPU comprising shelf ID, slot ID, CPU ID and rootport number. For example, shelf x, slot x, CPU x, rootport x |
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| Virtual Function Location | Location of virtual function comprising shelf ID, slot ID, device number, function number. For example, shelf x, slot x, device x, and vf x |

4.3 createIODeviceFunctions

Creates device functions specified by `functionCount`.

Syntax

```
#createIODeviceFunctions device address< IO device address> functionCount  
<number of device functions> rackID <Rack number> ShelfID <Shelf number>
```

Table 4-4 Parameters of createIODeviceFunctions

| Argument | Description |
|----------------------------|--|
| IO device address | Device address comprising shelf id, slot id, device id. For example, shelf x, slot y, device z. |
| number of device functions | The number of device functions to be created for the device. |
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |

4.4 destroyAllIODeviceFunctions

Destroys all the IO device functions.

Syntax

```
#destroyAllIODeviceFunctions deviceAddress <IO device address> rackID
<Rack number> ShelfID <Shelf number>
```

Table 4-5 Parameters of destroyAllIODeviceFunctions

| Argument | Description |
|-------------------|--|
| IO device address | Device address comprising shelf id, slot id, device id. For example, shelf x, slot y, device z. |
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |

4.5 eventLog

Gets the list of events available in the MaxCore.

Syntax

```
#eventLog rackID <Rack number> ShelfID <Shelf number>
```

Table 4-6 Parameters of eventLog

| Argument | Description |
|--------------|-------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |

4.6 getCPUAssignedFunctions

Gets the functions assigned to the PCI tree of CPU.

Syntax

```
#getCPUAssignedFunctions rackID <Rack number> ShelfID <Shelf number>  
slotID <physical slot number> cpuID <CPU number> rootPortNumber <root port  
number> [mode <running or updated configuration>]
```

Table 4-7 Parameters of getCPUAssignedFunctions

| Argument | Description |
|----------------------|-------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| physical slot number | Physical slot number of target CPU. |
| CPU number | CPU number. |
| root port number | Root port number of target CPU. |
| configuration mode | running or updated configuration. |

4.7 getCPUVirtualNICFunctionCount

Gets the number of virtual NIC functions of a CPU located by its address (slot number, CPU number).

Syntax

```
#getCPUVirtualNICFunctionCount cpuAddress <CPU address> rackID <Rack  
number> ShelfID <Shelf number>
```

Table 4-8 Parameters of getCPUVirtualNICFunctionCount

| Argument | Description |
|--------------|---|
| CPU address | Location of CPU comprising shelf ID, slot ID, CPU ID and rootport number. For example, shelf x, slot x, CPU x, rootport x. |
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |

4.8 getChassisPayloadPowerPolicy

Gets chassis power policy for a shelf. Payload power on/off.

Syntax

```
#getChassisPayloadPowerPolicy rackID <Rack number> ShelfID <Shelf number>
```

Table 4-9 Parameters of getChassisPayloadPowerPolicy

| Argument | Description |
|--------------|-------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |

4.9 getCoolingPolicy

Gets the shelf cooling policy - low noise or high reliability.

Syntax

```
#getCoolingPolicy rackID <Rack number> ShelfID <Shelf number>
```

Table 4-10 Parameters of getCoolingPolicy

| Argument | Description |
|--------------|--|
| CPU address | Location of CPU comprising shelf ID, slot ID, CPU ID and rootport number. For example, shelf x, slot x, CPU x, rootport x |
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |

4.10 getFunctionAssignedCPUAddress

Gets the CPU address to which the Virtual Function (VF) is assigned.

Syntax

```
#getFunctionAssignedCPUAddress Rack ID <Rack number> ShelfID <Shelf number> vfLocation <virtual function location>
```

Table 4-11 Parameters of getFunctionAssignedCPUAddress

| Argument | Description |
|--------------|-------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |

getPEXMode

Table 4-11 Parameters of getFunctionAssignedCPUAddress (continued)

| Argument | Description |
|---------------------------|--|
| virtual function location | Location of virtual function comprising shelf ID, slot ID, device number, function number. For example, shelf x, slot x, device x, vf x |

4.11 getPEXMode

Gets PEX mode to basic or express fabric mode.

Syntax

```
#getPEXMode rackID <Rack number> ShelfID <Shelf number>
```

Table 4-12 Parameters of getPEXMode

| Argument | Description |
|--------------|-------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |

4.12 getSATAAssignment

Gets SATA assignment to shelf hosts. For example, SATA 1 and 2 assigned to slot 1, cpu 1 SATA 3 and 4 assigned to slot 1, cpu 2.

Syntax

```
#getSATAAssignment rackID <Rack number> ShelfID <Shelf number>
```

Table 4-13 Parameters of getSATAAssignment

| Argument | Description |
|--------------|-------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |

4.13 getShelfHostAddress

Gets the address of shelf host (slot number, CPU number).

Syntax

```
#getShelfHostAddress rackID <Rack number> ShelfID <Shelf number>
```

Table 4-14 Parameters of getShelfHostAddress

| Argument | Description |
|--------------|-------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |

4.14 getUSBAssignment

Gets USB assignment of shelf slots.

Syntax

```
#getUSBAssignment rackID <Rack number> ShelfID <Shelf number>
```

Table 4-15 Parameters of getUSBAssignment

| Argument | Description |
|--------------|-------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |

4.15 getZoneType

Gets the zone type - host or downstream.

Syntax

```
#getZoneType rackID <Rack number> ShelfID <Shelf number> slotID <slot number>
```

Table 4-16 Parameters of getZoneType

| Argument | Description |
|--------------|-------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| slot number | Displays physical slot number. |

4.16 listAlarms

Lists of alarms - Critical/Major/Minor.

Syntax

```
#getUSBAssignment rackID <Rack number> ShelfID <Shelf number>
```

Table 4-17 Parameters of listAlarms

| Argument | Description |
|--------------|-------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |

4.17 listAvailableCPUs

Provides list of all the available CPUs in the MaxCore.

Syntax

```
#listAvailableCPU rackID <Rack number> ShelfID <Shelf number>
```

Table 4-18 Parameters of listAvailableCPUs

| Argument | Description |
|--------------|-------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |

4.18 listAvailableIODevices

Provides list the available IO devices in the MaxCore.

Syntax

```
#listAvailableIODevices rackID <Rack number> ShelfID <Shelf number>
```

Table 4-19 Parameters of listAvailableIODevices

| Argument | Description |
|--------------|-------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |

4.19 listShelfInfo

Lists info of all the available shelves/MaxCore in the system.

Syntax

```
#listShelfInfo rackID <Rack number> ShelfID <Shelf number>
```

Table 4-20 Parameters of listShelfInfo

| Argument | Description |
|--------------|-------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |

4.20 removeIODeviceFunctionFromCPU

Removes the function of an I/O device from the PCI tree of a CPU.

Syntax

```
#removeIODeviceFunctionFromCPU cpuAddress <cpu address> rackID <Rack number> ShelfID <Shelf number> vfLocation <virtual function>
```

Table 4-21 Parameters of removeIODeviceFunctionFromCPU

| Argument | Description |
|---------------------------|--|
| CPU address | Location of CPU comprising shelf ID, slot ID, CPU ID and rootport number. For example, shelf x, slot x, CPU x, rootport x |
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| virtual function location | Location of virtual function comprising shelf ID, slot ID, device number, function number. For example, shelf x, slot x, device x, vf x |

4.21 resetPEXConfiguration

This method resets the PEX configuration to last running configuration.

Syntax

```
#resetPEXConfiguration rackID <Rack number> ShelfID <Shelf number>
```

Table 4-22 Parameters of resetPEXConfiguration

| Argument | Description |
|--------------|-------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |

4.22 setCPUVirtualNICFunctionCount

Sets the number of virtual NIC functions of a CPU located by its address (slot number, CPU number).

Syntax

```
#setCPUVirtualNICFunctionCount cpuAddress <cpu address> functionCount  
<virtual function> rackID <Rack number> ShelfID <Shelf number>
```

Table 4-23 Parameters of setCPUVirtualNICFunctionCount

| Argument | Description |
|----------------|--|
| CPU address | Location of CPU comprising shelf ID, slot ID, CPU ID and rootport number. For example, shelf x, slot x, CPU x, rootport x |
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| function count | Virtual NIC function count. |

4.23 setChassisPayloadPowerPolicy

Sets chassis power policy for a shelf. Power on/off payload.

Syntax

```
#setChassisPayloadPowerPolicy chassisPowerPolicy <chassis Power Policy>
rackID <rack number> shelfID <shelf number>
```

Table 4-24 Parameters of removeIODeviceFunctionFromCPU

| Argument | Description |
|----------------------|--|
| chassis Power Policy | Set power policy for a shelf. Power on/off payload when the shelf is powered on. |
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |

4.24 setCoolingPolicy

Sets the shelf cooling policy.

Syntax

```
#setCoolingPolicy coolingPolicy <cooling policy> rackID <Rack number>
ShelfID <Shelf number>
```

Table 4-25 Parameters of setCoolingPolicy

| Argument | Description |
|----------------|---|
| cooling policy | Cooling policy of a shelf - high reliability or low noise |
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |

4.25 setPEXMode

Sets PEX mode to basic or express fabric mode.

Syntax

```
#setPEXMode mode <PEX mode> rackID <Rack number> ShelfID <Shelf number>
```

Table 4-26 Parameters of setPEXMode

| Argument | Description |
|--------------|--|
| mode | PEX mode - base mode or express fabric mode. |
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |

4.26 setSATAAssignment

Sets SATA assignment to shelf hosts. For example: SATA 1 and 2 assigned to slot 1, CPU 1 SATA 3 and 4 assigned to slot 1, CPU 2.

Syntax

```
#setSATAAssignment rackID <Rack number> sataAssignment <sata assignment>  
ShelfID <Shelf number>
```

Table 4-27 Parameters of setSATAAssignment

| Argument | Description |
|-----------------|---|
| Rack number | Rack number of the target chassis. |
| sata assignment | SATA assignment for slot 1/15 comprising the CPU addresses and disk numbers. For example, {slot 1,cpu 2,disk 4};{slot 15,cpu 1,disk 1};{slot 1,cpu 2,disk 2};{slot 15,cpu 1,disk 3}; |
| Shelf number | Shelf number of the target chassis. |

4.27 setUSBAssignment

Sets USB assignment of shelf slots.

Syntax

```
#setUSBAssignment rackID <Rack number> ShelfID <Shelf number>
usbAssignment <USB assignment>
```

Table 4-28 Parameters of setUSBAssignment

| Argument | Description |
|----------------|--|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| USB assignment | USB assigned CPU addresses separated by commas. For example, Each USB Assignment entry is as follows - (Slot X, CPU Y, USB Z) |

4.28 getBIOSBootOrder

Gets the value of BIOS boot parameters related to boot order configuration of CPU located by its address (slot number, CPU number).

Syntax

```
#getBIOSBootOrder rackID <Rack number> shelfID <Shelf number> slotID <Slot
number> cpuID <CPU number>
```

Table 4-29 Parameters of getBIOSBootOrder

| Argument | Description |
|--------------|-------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| Slot number | Slot number of target CPU. |
| CPU number | CPU number. |

4.29 getBIOSConsoleRedirectionParameters

Gets the value of BIOS boot parameters related to Console Redirection of CPU located by its address (slot number, CPU number).

Syntax

```
#getBIOSConsoleRedirectionParameters rackID <Rack number> shelfID <Shelf number> slotID <Slot number> cpuID <CPU number>
```

Table 4-30 Parameters of getBIOSConsoleRedirectionParameters

| Argument | Description |
|--------------|-------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| Slot number | Slot number of target CPU. |
| CPU number | CPU number. |

4.30 getBIOSBootConfiguration

Gets the value of BIOS boot parameters related to Boot Configuration of CPU located by its address (slot number, CPU number).

Syntax

```
#getBIOSBootConfiguration rackID <Rack number> shelfID <Shelf number> slotID <Slot number> cpuID <CPU number>
```

Table 4-31 Parameters of getBIOSBootConfiguration

| Argument | Description |
|--------------|-------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| Slot number | Slot number of target CPU. |
| CPU number | CPU number. |

4.31 getBIOSPeripheralConfiguration

Gets the value of BIOS boot parameters related to Peripheral configuration of CPU located by its address (slot number, CPU number).

Syntax

```
#getBIOSPeripheralConfiguration rackID <Rack number> shelfID <Shelf number> slotID <Slot number> cpuID <CPU number>
```

Table 4-32 Parameters of getBIOSPeripheralConfiguration

| Argument | Description |
|--------------|-------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| Slot number | Slot number of target CPU. |
| CPU number | CPU number. |

4.32 getBIOSIIIOConfiguration

Gets the value of BIOS boot parameters related to IIO configuration of CPU located by its address (slot number, CPU number).

Syntax

```
#getBIOSIIIOConfiguration rackID <Rack number> shelfID <Shelf number> slotID <Slot number> cpuID <CPU number>
```

Table 4-33 Parameters of getBIOSIIIOConfiguration

| Argument | Description |
|--------------|-------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| Slot number | Slot number of target CPU. |
| CPU number | CPU number. |

4.33 getBIOSSATAConfiguration

Gets the value of BIOS boot parameters related to PCH SATA configuration of CPU located by its address (slot number, CPU number).

Syntax

```
#getBIOSSATAConfiguration rackID <Rack number> shelfID <Shelf number>  
slotID <Slot number> cpuID <CPU number>
```

Table 4-34 Parameters of getBIOSSATAConfiguration

| Argument | Description |
|--------------|-------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| Slot number | Slot number of target CPU. |
| CPU number | CPU number. |

4.34 getBIOSVideoConfiguration

Gets the value of BIOS boot parameters related to Video configuration of CPU located by its address (slot number, CPU number).

Syntax

```
#getBIOSVideoConfiguration rackID <Rack number> shelfID <Shelf number>  
slotID <Slot number> cpuID <CPU number>
```

Table 4-35 Parameters of getBIOSVideoConfiguration

| Argument | Description |
|--------------|-------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| Slot number | Slot number of target CPU. |
| CPU number | CPU number. |

4.35 getBIOSUSBConfiguration

Gets the value of BIOS boot parameters related to USB configuration of CPU located by its address (slot number, CPU number).

Syntax

```
#getBIOSUSBConfiguration rackID <Rack number> shelfID <Shelf number>
slotID <Slot number> cpuID <CPU number>
```

Table 4-36 Parameters of getBIOSUSBConfiguration

| Argument | Description |
|--------------|-------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| Slot number | Slot number of target CPU. |
| CPU number | CPU number. |

4.36 getBIOSProcessorConfiguration

Gets the value of BIOS boot parameters related to Processor configuration of CPU located by its address (slot number, CPU number).

Syntax

```
#getBIOSProcessorConfiguration rackID <Rack number> shelfID <Shelf
number> slotID <Slot number> cpuID <CPU number>
```

Table 4-37 Parameters of getBIOSProcessorConfiguration

| Argument | Description |
|--------------|-------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| Slot number | Slot number of target CPU. |
| CPU number | CPU number. |

4.37 getBIOSPowerManagementConfiguration

Gets the value of BIOS boot parameters related to Power Management configuration of CPU located by its address (slot number, CPU number).

Syntax

```
#getBIOSPowerManagementConfiguration rackID <Rack number> shelfID <Shelf number> slotID <Slot number> cpuID <CPU number>
```

Table 4-38 Parameters of getBIOSPowerManagementConfiguration

| Argument | Description |
|--------------|-------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| Slot number | Slot number of target CPU. |
| CPU number | CPU number. |

4.38 getBIOSThermalConfiguration

Gets the value of BIOS boot parameters related to Thermal Configuration of CPU located by its address (slot number, CPU number).

Syntax

```
#getBIOSThermalConfiguration rackID <Rack number> shelfID <Shelf number> slotID <Slot number> cpuID <CPU number>
```

Table 4-39 Parameters of getBIOSThermalConfiguration

| Argument | Description |
|--------------|-------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| Slot number | Slot number of target CPU. |
| CPU number | CPU number. |

4.39 getBIOSMemoryConfiguration

Gets the value of BIOS boot parameters related to Memory Configuration of CPU located by its address (slot number, CPU number).

Syntax

```
#getBIOSMemoryConfiguration rackID <Rack number> shelfID <Shelf number>
slotID <Slot number> cpuID <CPU number>
```

Table 4-40 Parameters of getBIOSMemoryConfiguration

| Argument | Description |
|--------------|-------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| Slot number | Slot number of target CPU. |
| CPU number | CPU number. |

4.40 getBIOSCommonRefcodeConfiguration

Gets the value of BIOS boot parameters related to Common Refcode Configuration of CPU located by its address (slot number, CPU number).

Syntax

```
#getBIOSCommonRefcodeConfiguration rackID <Rack number> shelfID <Shelf
number> slotID <Slot number> cpuID <CPU number>
```

Table 4-41 Parameters of getBIOSCommonRefcodeConfiguration

| Argument | Description |
|--------------|-------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| Slot number | Slot number of target CPU. |
| CPU number | CPU number. |

4.41 setBIOSBootOrderNew

Replaces the current boot order configuration with the fresh one of CPU located by its address (slot number, CPU number).

Syntax

```
#setBIOSBootOrderNew rackID <Rack number> shelfID <Shelf number> slotID
<Slot number> cpuID <CPU number> BootOrder <Boot Order>
```

Table 4-42 Parameters of setBIOSBootOrderNew

| Argument | Description |
|--------------|--|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| Slot number | Slot number of target CPU. |
| CPU number | CPU number. |
| BootOrder | Boot Order: Boot Order: [< bootdev > {[, < bootdev >]}] bootdev= full name of a boot option, as it is shown by the BIOS, or one of the following predefined selfspeaking names: {sdcard/efisdcard/bpusb/bpusb0/bpusb1/bpusb2/bpusb3/efibpusb/efibpusb0/efibpusb1/efibpusb2/efibpusb3/iomusb/iomusb0/iomusb1/iomusb2/iomusb3/efiiomusb/efiiomusb0/efiiomusb1/efiiomusb2/efiiomusb3/iomsata0/iomsata1/efiiomsata0/efiiomsata1/efishell} For example, sdcard,efisdcard. |
| Result | Value of the selected parameter. |

NOTE: When configuring boot order for an EFI boot device, mention the full name of the boot option (as it is shown by the BIOS) in the BootOrder argument. For example:

```
#setBIOSBootOrderNew rackID 1 shelfID 1 slotID 1 cpuID 1 BootOrder "EFI
Net 0 IPv4 (02-0B-00-10-02-02),sdcard".
```

4.42 setBIOSBootOrderDefault

Sets the boot order configuration back to default of CPU located by its address (slot number, CPU number).

Syntax

```
#setBIOSBootOrderDefault rackID <Rack number> shelfID <Shelf number>
slotID <Slot number> cpuID <CPU number>
```

Table 4-43 Parameters of setBIOSBootOrderDefault

| Argument | Description |
|--------------|-------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| Slot number | Slot number of target CPU. |
| CPU number | CPU number. |
| Result | Value of the selected parameter. |

4.43 setBIOSBootOrderbyIndex

Modifies the existing boot order configuration according to index and boot device provided of CPU located by its address (slot number, CPU number).

Syntax

```
#setBIOSBootOrderbyIndex rackID <Rack number> shelfID <Shelf number>
slotID <Slot number> cpuID <CPU number> BootDevice <Boot Device> Index
<Index>
```

Table 4-44 Parameters of setBIOSBootOrderbyIndex

| Argument | Description |
|--------------|--|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| Slot number | Slot number of target CPU. |
| CPU number | CPU number. |
| Boot Device | Boot Device: full name of a boot option, as it is shown by the BIOS, or one of the following predefined selfspeaking names: {sdcard/efisdcard/bpusb/bpusb0/bpusb1/bpusb2/bpusb3/efibpusb/efibpusb0/efi bpusb1/efibpusb2/efibpusb3/iomusb/iomusb0/iomusb1/iomusb2/iomusb3/efi iomusb/efiiomusb0/efiiomusb1/efiiomusb2/efiiomusb3/iomsata0/iomsata1/efiiom sata0/efiiomsata1/efishell} |
| Index | Number representing position in boot order. Possible values are 1,2,3 etc. Lower number represents higher priority. |
| Result | Value of the selected parameter. |

setBIOSConsoleRedirectionParameters

NOTE: When configuring boot order for an EFI boot device, mention the full name of the boot option (as it is shown by the BIOS) in the `BootDevice` argument. For example:

```
#setBIOSBootOrderNew rackID 1 shelfID 1 slotID 1 cpuID 1 Index 1
BootDevice "EFI Net 0 IPv4 (02-0B-00-10-02-02)".
```

4.44 setBIOSConsoleRedirectionParameters

Sets the value of BIOS boot parameters related to Console Redirection of CPU located by its address (slot number, CPU number).

Syntax

```
#setBIOSConsoleRedirectionParameters rackID <Rack number> shelfID <Shelf
number> slotID <Slot number> cpuID <CPU number> BaudRate <Baud Rate>
```

Table 4-45 Parameters of setBIOSConsoleRedirectionParameters

| Argument | Description |
|--------------|--|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| Slot number | Slot number of target CPU. |
| CPU number | CPU number. |
| BaudRate | Baud Rate: 9600(0), 19200(1), 38400(2), 57600(3), 115200(4). |

4.45 setBIOSBootConfiguration

Sets the value of BIOS boot parameters related to Boot Configuration of CPU located by its address (slot number, CPU number).

Syntax

```
#setBIOSBootConfiguration rackID <Rack number> shelfID <Shelf number>
slotID <Slot number> cpuID <CPU number> SkipShell <Skip Shell> AutoReset
<Auto Reset> BootType <Boot Type> NetworkStack <Network Stack>
PXEBootCapability <PXE Boot Capability> USBBoot <USB boot>
```

Table 4-46 Parameters of setBIOSBootConfiguration

| Argument | Description |
|-------------|------------------------------------|
| Rack number | Rack number of the target chassis. |

Table 4-46 Parameters of setBIOSBootConfiguration (continued)

| Argument | Description |
|-------------------|---|
| Shelf number | Shelf number of the target chassis. |
| Slot number | Slot number of target CPU. |
| CPU number | CPU number. |
| SkipShell | Skip Internal EFI Shell: OFF(1), ON(0). |
| AutoReset | Auto Reset: OFF(1), ON(0). |
| BootType | Boot Type: dual(0), legacy(1), uefi(2). |
| NetworkStack | Network Stack: OFF(1), ON(0). |
| PXEBootCapability | PXE Boot Capability: ipv4(0), ipv6(1), ipv4v6(2). |
| USBBoot | USB Boot: OFF(1), ON(0). |

4.46 setBIOSPeripheralConfiguration

Sets the value of BIOS boot parameters related to Peripheral configuration of CPU located by its address (slot number, CPU number).

Syntax

```
#setBIOSPeripheralConfiguration rackID <Rack number> shelfID <Shelf
number> slotID <Slot number> cpuID <CPU number> PCIeSRIOV <PCIe SR-IOV>
PCIeARI <PCIe ARI> Pci64bitDecode <Pci 64-bit Decode> SpreadSpectrum
<Spread Spectrum>
```

Table 4-47 Parameters of setBIOSPeripheralConfiguration

| Argument | Description |
|----------------|-------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| Slot number | Slot number of target CPU. |
| CPU number | CPU number. |
| PCIeSRIOV | PCIe SR-IOV: OFF(1), ON(0). |
| PCIeARI | PCIe ARI: OFF(1), ON(0). |
| Pci64bitDecode | Pci 64-bit Decode: OFF(1), ON(0). |

setBIOSIIOConfiguration

Table 4-47 Parameters of setBIOSPeripheralConfiguration (continued)

| Argument | Description |
|----------------|---------------------------------|
| SpreadSpectrum | Spread Spectrum: OFF(1), ON(0). |
| Result | Result of the operation. |

4.47 setBIOSIIOConfiguration

Sets the value of BIOS boot parameters related to IIO configuration of CPU located by its address (slot number, CPU number).

Syntax

```
#setBIOSIIOConfiguration rackID <Rack number> shelfID <Shelf number>  
slotID <Slot number> cpuID <CPU number> VTd <VT d> InterruptRemapping  
<Interrupt Remapping>
```

Table 4-48 Parameters of setBIOSIIOConfiguration

| Argument | Description |
|--------------------|--|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| Slot number | Slot number of target CPU. |
| CPU number | CPU number. |
| VTd | Intel VT for Directed I/O (VT-d): OFF(1), ON(0). |
| InterruptRemapping | Interrupt Remapping: OFF(1), ON(0). |
| Result | Result of the operation. |

4.48 setBIOS SATA Configuration

Sets the value of BIOS boot parameters related to PCH SATA configuration of CPU located by its address (slot number, CPU number).

Syntax

```
#setBIOS SATA Configuration rackID <Rack number> shelfID <Shelf number>
slotID <Slot number> cpuID <CPU number> SATAController <SATA Controller>
SATAMode <SATA Mode> RAIDOROMdelay <RAID OROM delay> ALPMSupport <ALPM
Support> SATASpeed <SATA Speed>
```

Table 4-49 Parameters of setBIOS SATA Configuration

| Argument | Description |
|----------------|--|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| Slot number | Slot number of target CPU. |
| CPU number | CPU number. |
| SATAController | SATA Controller: OFF(1), ON(0). |
| SATAMode | HDC configured as (SATA Mode Selection): ide(0), ahci(1), raid(2). |
| RAIDOROMdelay | RAID OROM prompt delay (in sec): 2(0), 4(1), 6(2). |
| ALPMSupport | Aggressive LPM Support: OFF(1), ON(0). |
| SATASpeed | SATA Speed Support (in Gb/s): 1.5(0), 3(1), 6(2). |
| Result | Result of the operation. |

4.49 setBIOS Video Configuration

Sets the value of BIOS boot parameters related to Video configuration of CPU located by its address (slot number, CPU number).

Syntax

```
#setBIOS Video Configuration rackID <Rack number> shelfID <Shelf number>
slotID <Slot number> cpuID <CPU number> DisplayMode <Display Mode>
```

setBIOSUSBConfiguration

Table 4-50 Parameters of getBIOSVideoConfiguration

| Argument | Description |
|--------------|---|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| Slot number | Slot number of target CPU. |
| CPU number | CPU number. |
| DisplayMode | Display Mode: usual(0), ignoreoromio[Ignore VGA Option ROM and I/O Space](1). |
| Result | Result of the operation. |

4.50 setBIOSUSBConfiguration

Sets the value of BIOS boot parameters related to USB configuration of CPU located by its address (slot number, CPU number).

Syntax

```
#setBIOSUSBConfiguration rackID <Rack number> shelfID <Shelf number>  
slotID <Slot number> cpuID <CPU number> USBBIOSsupport <USB BIOS Support>  
UsbPort0 <Usb port0,1,2...7>
```

Table 4-51 Parameters of setBIOSUSBConfiguration

| Argument | Description |
|----------------|-------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| Slot number | Slot number of target CPU. |
| CPU number | CPU number. |
| USBBIOSsupport | USB BIOS Support: OFF(1), ON(0). |
| UsbPort0 | USB Port 0: OFF(1), ON(0). |
| UsbPort1 | USB Port 1: OFF(1), ON(0). |
| UsbPort2 | USB Port 2: OFF(1), ON(0). |
| UsbPort3 | USB Port 3: OFF(1), ON(0). |
| UsbPort4 | USB Port 4: OFF(1), ON(0). |

Table 4-51 Parameters of setBIOSUSBConfiguration (continued)

| Argument | Description |
|----------|----------------------------|
| UsbPort5 | USB Port 5: OFF(1), ON(0). |
| UsbPort6 | USB Port 6: OFF(1), ON(0). |
| UsbPort7 | USB Port 7: OFF(1), ON(0). |
| Result | Result of the operation. |

4.51 setBIOSProcessorConfiguration

Sets the value of BIOS boot parameters related to Processor configuration of CPU located by its address (slot number, CPU number).

Syntax

```
#setBIOSProcessorConfiguration rackID <Rack number> shelfID <Shelf number> slotID <Slot number> cpuID <CPU number> HyperThreading <HyperThreading> PerformanceWatt <Performance Watt> ExecuteDisableBit <ExecuteDisableBit> EnableLTSX <Enable LTSX> VMX <VMX> HardwarePrefetcher <HardwarePrefetcher> AdjCachePrefetch <AdjCachePrefetch> DCA <DCA> X2APIC <X2APIC>
```

Table 4-52 Parameters of setBIOSProcessorConfiguration

| Argument | Description |
|--------------------|---|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| Slot number | Slot number of target CPU. |
| CPU number | CPU number. |
| HyperThreading | Hyper-Threading [ALL]: OFF(1), ON(0). |
| PerformanceWatt | Performance/Watt: Traditional(0), Power Optimized(1). |
| ExecuteDisableBit | Execute Disable Bit: OFF(1), ON(0). |
| EnableLTSX | Enable LTSX: OFF(1), ON(0). |
| VMX | VMX: OFF(1), ON(0). |
| HardwarePrefetcher | Hardware Prefetcher: OFF(1), ON(0). |
| AdjCachePrefetch | Adjacent Cache Prefetch: OFF(1), ON(0). |

setBIOSPowerManagementConfiguration

Table 4-52 Parameters of setBIOSProcessorConfiguration (continued)

| Argument | Description |
|----------|--|
| DCA | Direct Cache Access: OFF(0), ON(1), AUTO(2). |
| X2APIC | X2APIC: OFF(1), ON(0). |
| Result | Result of the operation. |

4.52 setBIOSPowerManagementConfiguration

Sets the value of BIOS boot parameters related to Power Management configuration of CPU located by its address (slot number, CPU number).

Syntax

```
#setBIOSPowerManagementConfiguration rackID <Rack number> shelfID <Shelf number> slotID <Slot number> cpuID <CPU number> EIST <EIST> TurboMode <Turbo Mode> CPUCState <CPU C State> PackageCStatelimit <Package C State limit> CPUC3report <CPU C3 report> CPUC6report <CPU C6 report> EnhancedHaltState <Enhanced Halt State> OSACPICx <OS ACPI Cx>
```

Table 4-53 Parameters of setBIOSPowerManagementConfiguration

| Argument | Description |
|--------------------|---|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| Slot number | Slot number of target CPU. |
| CPU number | CPU number. |
| EIST | EIST (GV3): OFF(1), ON(0). |
| TurboMode | Turbo Mode: OFF(1), ON(0). |
| CPUCState | CPU C State: OFF(1), ON(0). |
| PackageCStatelimit | Package C State limit: c0c1(0), c2(1), c6nr(2). |
| CPUC3report | CPU C3 report: OFF(1), ON(0). |
| CPUC6report | CPU C6 report: OFF(1), ON(0). |
| EnhancedHaltState | Enhanced Halt State (C1E): OFF(1), ON(0). |
| OSACPICx | OS ACPI Cx: c2(0), c3(1). |
| Result | Result of the operation. |

4.53 setBIOSThermalConfiguration

Sets the value of BIOS boot parameters related to Thermal Configuration of CPU located by its address (slot number, CPU number).

Syntax

```
#setBIOSThermalConfiguration rackID <Rack number> shelfID <Shelf number>
slotID <Slot number> cpuID <CPU number> DTS <DTS>
```

Table 4-54 Parameters of setBIOSThermalConfiguration

| Argument | Description |
|--------------|-------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| Slot number | Slot number of target CPU. |
| CPU number | CPU number. |
| DTS | DTS: OFF(1), ON(0). |
| Result | Result of the operation. |

4.54 setBIOSMemoryConfiguration

Sets the value of BIOS boot parameters related to Memory Configuration of CPU located by its address (slot number, CPU number).

Syntax

```
#setBIOSMemoryConfiguration rackID <Rack number> shelfID <Shelf number>
slotID <Slot number> cpuID <CPU number> MemoryFrequency <Memory Frequency>
HaltOnMemTrainError <Halt on mem Training Error>
```

Table 4-55 Parameters of setBIOSMemoryConfiguration

| Argument | Description |
|---------------------|--|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| Slot number | Slot number of target CPU. |
| CPU number | CPU number. |
| MemoryFrequency | Memory Frequency: auto(0), 1333(1), 1600(2), 1867(3), 2133(4). |
| HaltOnMemTrainError | Halt on mem Training Error: OFF(1), ON(0). |

setBIOSCommonRefcodeConfiguration

Table 4-55 Parameters of setBIOSMemoryConfiguration (continued)

| Argument | Description |
|----------|--------------------------|
| Result | Result of the operation. |

4.55 setBIOSCommonRefcodeConfiguration

Sets the value of BIOS boot parameters related to Common Refcode Configuration of CPU located by its address (slot number, CPU number).

Syntax

```
#setBIOSCommonRefcodeConfiguration rackID <Rack number> shelfID <Shelf number> slotID <Slot number> cpuID <CPU number> Numa <Numa>
```

Table 4-56 Parameters of setBIOSCommonRefcodeConfiguration

| Argument | Description |
|--------------|-------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| Slot number | Slot number of target CPU. |
| CPU number | CPU number. |
| Numa | Numa: OFF(1), ON(0). |
| Result | Result of the operation. |

4.56 powerOffAllApplicationCPUs

It will Initiate the graceful shutdown or poweroff all application CPUs.

Syntax

```
#powerOffAllApplicationCPUs rackID <Rack number> shelfID <Shelf number> powerDown <Power> IncludeNetworkCPUs <Network CPUs> whence <whence>
```

Table 4-57 Parameters of powerOffAllApplicationCPUs

| Argument | Description |
|--------------|-------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |

Table 4-57 Parameters of powerOffAllApplicationCPUs (continued)

| Argument | Description |
|--------------------|--|
| IncludeNetworkCPUs | Include network CPUs like 9205 or not for power off. FALSE (o) or TRUE(1). |
| whence | If "atonce", power for CPU is set immediately. If "atstart", given power for CPU is applied at the start of Shelf Host. atonce (o) or atstart (1). |
| PowerDown | HARD_POWER_DOWN(0) or GRACEFUL_POWER_DOWN(1). |
| result | Result of the operation. |

4.57 setShelfAddress

It sets the new Shelf address, it contains new Shelf Name and ShelfId.

NOTICE

Before changing shelfid, you need to shutdown all application CPUs because system will go for power cycle. Shelf name changes is not required for shutdown of application CPUs. For more details, refer to [powerOffAllApplicationCPUs](#)

Syntax

```
#setShelfAddress rackID <Rack number> currentShelfID <Current shelf number> newShelfID <New shelf number> shelfName <New shelf name>
```

Table 4-58 Parameters of setShelfAddress

| Argument | Description |
|----------------------|---|
| Rack number | Rack number of the target chassis. |
| Current Shelf number | Current ShelfID |
| New shelf number | Provide the new ShelfID, its Range:1-255. |
| New shelf name | Provide the shelf name. |
| result | Result of the operation. |

4.58 getShelfAddress

It gets the Shelf address of required shelf, it contains name of Shelf and ShelfId.

Syntax

```
#getShelfAddress rackID <Rack number> shelfID <Shelf number>
```

Table 4-59 Parameters of getShelfAddress

| Argument | Description |
|--------------|-------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| result | Result of the operation. |

4.59 setBMCIPAddress

It will set the BMC IP Address, it contains BMC IP, Subnet Mask, Gateway IP.

Syntax

```
#setBMCIPAddress rackID <Rack number> shelfID <Shelf number>  
DHCPConfiguration <DHCP Configuration> BmcIPAddr <BMC IP Address>  
GatewayIPAddr <Gateway IP Address> SubnetMask <Subnet Mask>
```

Table 4-60 Parameters of setBMCIPAddress

| Argument | Description |
|--------------------|---|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| DHCP Configuration | DHCP Configuration: DISABLE (0) / ENABLE (1). |
| BMC IP Address | Provide the BMCIPAddr to set. |
| Gateway IP Address | Provide the Gateway IP to set. |
| Subnet Mask | Provide the SubnetMask to set. |
| result | Result of the operation. |

4.60 getBMCIPAddress

It gets the BMC IP Address, it contains BMC IP, Subnet Mask, Gateway IP and DHCP configuration.

Syntax

```
#getBMCIPAddress rackID <Rack number> shelfID <Shelf number>
```

Table 4-61 Parameters of getBMCIPAddress

| Argument | Description |
|--------------|-------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| result | Result of the operation. |

4.61 listShelves

It provides the list of all configured shelf in `maxcore.conf`.

Syntax

```
#listShelves
```

4.62 addShelf

It configures the new shelf for Multishelf.

NOTE: You need to Restart the `ssfCore` using `systemctl restart ssfCore.service` to effect the changes.

Syntax

```
#addShelf shelfID <Shelf number> shelfHostIpAddr <Shelf Host IP Address>  
master <master> shelfName <New shelf name>
```

Table 4-62 Parameters of addShelf

| Argument | Description |
|-----------------------|--|
| Shelf number | Shelf number of the new shelf. |
| Shelf Host IP Address | ShelfHost IP address of new Shelf on which SsfCore can connect. |
| master | Whether configured shelf is a master or a slave, for master true and for slave false |

removeShelf

Table 4-62 Parameters of addShelf (continued)

| Argument | Description |
|----------------|--|
| New shelf name | Provide ShelfName of the new shelf. Default name is Shelf. Note: Shelf Name limit is 16 characters. |
| result | Result of the operation. |

4.63 removeShelf

It removes the configured shelf.

NOTE: You need to restart the ssfCore using `systemctl restart ssfCore.service` to effect the changes.

Syntax

```
#removeShelf shelfID <Shelf number>
```

Table 4-63 Parameters of removeShelf

| Argument | Description |
|--------------|---|
| Shelf number | It ShelfID of Shelf which you want to remove. |
| result | Result of the operation. |

4.64 modifyShelf

It modifies the existing shelf data.

Note: You need to Restart the ssfCore using `systemctl restart ssfCore.service` to effect the changes.

Syntax

```
#modifyShelf currentShelfID <Current shelf number> newShelfID <New shelf number> newShelfHostIpAddr <New Shelf Host IP Address> isMaster <Master> newShelfName <New shelf name>
```

Table 4-64 Parameters of modifyShelf

| Argument | Description |
|----------------------|--|
| Current Shelf number | Current shelf number of the shelf. |
| New Shelf number | New shelf number of the modifiable shelf |

Table 4-64 Parameters of modifyShelf (continued)

| Argument | Description |
|---------------------------|--|
| New Shelf Host IP Address | New ShelfHost IP address of the shelf to which SsfCore can connect. |
| Master | Whether configured shelf is a master or a slave, for master true and for slave false . |
| New shelf name | New ShelfName of the shelf. |
| result | Result of the operation. |

4.65 setSsfHostInterfaceName

It sets the local ssfHost interface name to which ssfAgent can connect.

Syntax

```
#setSsfHostInterfaceName ssfHostInterface <SSF host interface name>
```

Table 4-65 Parameters of setSsfHostInterfaceName

| Argument | Description |
|-------------------------|--|
| SSF host interface name | Provide the name of local ssfHost interface name to which all ssfAgents can connect. |
| result | Result of the operation. |

4.66 getSsfHostInterfaceName

It gets the SSF host interface name (where ssfCore is running).

Syntax

```
#getSsfHostInterfaceName
```

4.67 setDefaultConfiguration

It sets the Virtual functions, disk and USB assignment to default.

Syntax

```
#setDefaultConfiguration rackID <Rack number> shelfID <Shelf number>
```

Table 4-66 Parameters of setDefaultConfiguration

| Argument | Description |
|--------------|-------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| result | Result of the operation. |

4.68 getDiscoveryStatus

It gets the status of basic services (like ipmitool, pciemgmt), and checking if shelf id is zero or not, and also Shelf discovery status.

Syntax

```
#getDiscoveryStatus rackID <Rack number> shelfID <Shelf number>
```

Table 4-67 Parameters of getDiscoveryStatus

| Argument | Description |
|--------------|-------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| result | Result of the operation. |

4.69 setPXEBootEnvironmentForCPU

It sets the configure PXE Boot for CPU. Also, it uploads boot images (kernel and ramdisk).

Syntax

```
#setPXEBootEnvironmentForCPU rackID <Rack number> shelfID <Shelf number>  
slotID <slot number> cpuID <CPU number> PXEBoot <Enable/Disable>  
uploadFile <True/False> filePath <absolute file path>
```

Table 4-68 Parameters of getDiscoveryStatus

| Argument | Description |
|--------------|--|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| Slot number | Slot number of target CPU. |
| CPU number | CPU number. |
| PXEBoot | Enable or Disable PXE Boot Configuration for CPU. |
| uploadFile | TRUE: for Upload file and FALSE for set the PXE BOOT Environment. |
| filePath | Provide absolute file path of boot images. For example (../pcie7410/filename). |
| result | Result of the operation. |

NOTE: You can use this command for uploading image set PXE Configuration and clear PXE configuration. For each operations, parameters and its values are mentioned in the below table.

| Operation | PXEBoot | uploadFile | filePath |
|-------------------------------------|---------|------------|-------------------|
| To upload the Image | DISABLE | TRUE | Absolute Filepath |
| To set the PXE Boot Configuration | ENABLE | FALSE | Optional |
| To clear the PXE Boot Configuration | DISABLE | FALSE | Optional |

4.70 getPXEBootConfigurationForCPU

It gets the PXE Boot Configuration for CPU (like kernel and ramdisk) image available or not.

Syntax

```
#getPXEBootConfigurationForCPU rackID <Rack number> shelfID <Shelf number> slotID <slot number> cpuID <CPU number>
```

Table 4-69 Parameters of getDiscoveryStatus

| Argument | Description |
|--------------|-------------------------------------|
| Rack number | Rack number of the target chassis. |
| Shelf number | Shelf number of the target chassis. |
| Slot number | Slot number of target CPU. |
| CPU number | CPU number. |
| result | Result of the operation. |

System Commands

System commands allows you to view detailed hardware and software configuration information of system. Hardware configuration details include Physical Execution Environments (PEE) and Virtual Execution Environments (VEE). Software configuration details include operating systems, services, software entities managed by VEE, and so on.

The system and its components information is displayed in the hierarchal tree structure. For example, the system information is displayed along with its child nodes, such as slots, and sites.

5.1 CLI Commands in disable mode

This section describes the CLI commands that are available in the disable mode, to enter the disable mode:

```
Trying ::1...
telnet: connect to address ::1: Connection refused
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Welcome to SSF CLI
Username: Admin
Password:
Access granted
>
```

Events display

The events display command is used to enable/disable the events that are received in the current session. If the events display is enabled the user can see the event messages on the console.

Syntax

```
>events display <mode>
```

For example:

```
>events display ?
  disable  Disable functionality
  enable   Enable functionality
```

set configuration:

The set configuration command is used to set the current configuration.

CLI commands in enable mode

Syntax

```
>set configuration <configuration file name>
```

The following table provides the list of Disable mode commands:

Table 5-1 Parameters of Disable mode commands

| Argument | Description |
|--------------------|--|
| actioninput | Enable or disable action input debugging |
| enable | Turn on privileged commands |
| events | Events related commands |
| exit | Exit from the EXEC |
| lexeroutput | Enable or disable lexer debug output |
| ping | Send echo messages |
| read-only-commands | Enable or disable showing commands with read-only access |
| reload | Reloads commands |
| shortcuts | Display shortcuts |
| show | Show running system information |
| telnet | Open a telnet connection |
| tracert | Trace route to destination. |

5.2 CLI commands in enable mode

The following procedure explains how to go to the enable mode:

```
Trying ::1...
telnet: connect to address ::1: Connection refused
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Welcome to SSF CLI
Username: Admin
Password:
Access granted
>enable
```

The following table provides the list of Enable mode commands:

Table 5-2 Parameters of Enable mode commands

| Argument | Description |
|-----------------|---|
| add | Create system object |
| assign | Keyword |
| configure | Enter configuration mode |
| copy | Copy a config script |
| debug | Configure debugging output |
| disable | Turn off privileged commands |
| event | Event configuration |
| exit | Exit from the EXEC |
| group | Keyword |
| halt | Shut down system |
| lock | Perform lock |
| logging | Configure logging |
| more | Configure more |
| no | Negate a command or set its defaults |
| password | Specify the password for the user |
| ping | Send echo messages |
| remove | Keyword |
| shell | The shell command executes a OS command |
| show | Show running system information |
| shutdown | Shut down system |
| snmp-server | Configure SNMP Agent |
| source | Execute script |
| telnet | Open a telnet connection |
| terminal | Set terminal line parameters |
| traceroute | Trace route to destination |

CLI commands in config mode

Table 5-2 Parameters of Enable mode commands (continued)

| Argument | Description |
|----------|---|
| username | Establish User Name Authentication |
| write | Save the current configuration to NVRAM |

5.3 CLI commands in config mode

The following procedure explains how enter into the config mode:

```
#Trying ::1...
telnet: connect to address ::1: Connection refused
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Welcome to SSF CLI
Username: Admin
Password:
Access granted
>enable
#configure terminal
```

CLI commands in system mode

```
deletelog
```

The deletelog command is used to clear the event log messages from the SSF system.

Syntax

```
#deletelog
```

eventFilterSeverity

The eventFilterSeverity command is used to store the logs based on the severity of events received in the system.

Event Severity is a 8 bit integer representing: 0 Disable (disable all logs), 1 Critical, 2 Major, 4 Minor, 8 Informational, 16 OK, 32 DEBUG, 255 All, It is possible to accumulate the values so that a combination of events can be filtered. Max allowed value is 255.

Syntax

```
# eventFilterSeverity <severity number>
```


eventFilterType

The eventFilterType command is used to store the events based on the type of the event. Event Type is a 16-bit integer representing:

```
0 Disable (disable all logs)
1  SSF_EVENT_TYPE_MASK_SENSOR
2  SSF_EVENT_TYPE_MASK_SENSORENABLE
4  SSF_EVENT_TYPE_MASK_WATCHDOG
8  SSF_EVENT_TYPE_MASK_OEM
16 SSF_EVENT_TYPE_MASK_SOFTWARE
32 SSF_EVENT_TYPE_MASK_HOTSWAP
64 SSF_EVENT_TYPE_MASK_RESOURCE
128 SSF_EVENT_TYPE_MASK_DOMAIN
256 SSF_EVENT_TYPE_MASK_USER
512 SSF_EVENT_TYPE_MASK_REGISTRATION
```

It is possible to accumulate the values so that a combination of events can be filtered. Max allowed value is 1023.

Syntax

```
# eventFilterType <event type number>
```

listlog

The listlog command is used to show the events in the system.

Syntax

```
# listlog
```

maxNoEvents

The maxNoEvents command is used to store maximum of events in the system.

Syntax

```
# maxNoEvents
```

userConfig

The userConfig command is used to allow the user to decide whether to apply the configuration or not.

User confirmation attribute for applying heterogeneous configurations.

0 - Yes, can apply the configuration.

CLI commands in config mode

1 - No, Do not apply the configuration

Syntax

```
# userConfig <config attribute>
```

logfilter

The logfilter command allows user to configure the log sinks.

Syntax

```
# logfilter <logsink name>
```

After entering logsink say syslog, user can configure the module and priority of that logsink.

modules

```
MaxCore(logfilter-syslog)#modules ?
```

A sequence of modules names separated by comma, for instance WEB,SYSTEM,CLI (applicable for module log filter) [empty by default]

```
MaxCore(logfilter-syslog)#priority ?
```

```
critical  Filter priority (applicable for priority log filter) [debug]
debug     Filter priority (applicable for priority log filter) [debug]
error     Filter priority (applicable for priority log filter) [debug]
info      Filter priority (applicable for priority log filter) [debug]
warning   Filter priority (applicable for priority log filter) [debug]
```

The following table provides the list of Configure mode commands:

Table 5-3 Parameters of Configure mode commands

| Argument | Description |
|--------------------|---|
| UsingMOIDQueryNode | MOIDQueryNode is used |
| event | The shell command executes a OS command |
| exit | Exit from configure mode |
| no | Negate a command or set its defaults |
| show | Show running system information |
| system | Configure system |
| logfilter | Configure logfilter |

5.4 CLI Commands in System Mode

All system commands are available in system mode. To use the system commands, you need to enter into the system mode.

Syntax

>enable

#configure terminal

MaxCore(config)#

MaxCore(config)#system 1

MaxCore(system-1)#?

The following table provides the list of system mode commands:

Table 5-4 Parameters of System mode commands

| Argument | Description |
|---------------------|--|
| eventFilterSeverity | Severity of the events to store. Event Severity is a 8 bit integer representing: 0 Disable (disable all logs), 1 Critical, 2 Major, 4 Minor, 8 Informational, 16 OK, 32 DEBUG, 255 All, It is possible to accumulate the values so that a combination of events can be filtered. Max allowed value is 255 |
| eventFilterType | Type of the events to store. Event Type is a 16 bit integer representing: 0 Disable (disable all logs), 1 SSF_EVENT_TYPE_MASK_SESNSOR, 2 SSF_EVENT_TYPE_MASK_SENSORENABLE, 4 SSF_EVENT_TYPE_MASK_WATCHDOG, 8 SSF_EVENT_TYPE_MASK_OEM, 16 SSF_EVENT_TYPE_MASK_SOFTWARE, 32 SSF_EVENT_TYPE_MASK_HOTSWAP, 64 SSF_EVENT_TYPE_MASK_RESOURCE, 128 SSF_EVENT_TYPE_MASK_DOMAIN, 256 SSF_EVENT_TYPE_MASK_USER, 512 SSF_EVENT_TYPE_MASK_REGISTRATION, It is possible to accumulate the values so that a combination of events can be filtered. Max allowed value is 1023 |
| exit | Exit from system configuration mode |
| maxNoEvents | Maximum number events that can be stored in system |
| no | Negate a command or set its defaults |
| shelf | Configure shelf |
| show | Show running system information |

Table 5-4 Parameters of System mode commands

| Argument | Description |
|-----------------|---|
| softwareVersion | To display Management software information. |
| systemInfo | Describes the node |
| systemName | Name of the System |
| userConfig | User confirmation attribute for applying heterogeneous configurations. 0 - Yes, you can apply the configuration. 1 - No, do not apply the configuration |

5.5 exit

The exit command is available at every mode of the system. This command allows you to exit from a particular mode.

Syntax

```
#exit
```

5.6 CLI Commands for Shelf Mode

All Shelf commands are available in shelf mode. To use the shelf commands, you need to enter into the shelf mode.

Syntax

```
>enable  
#configure terminal  
MaxCore(config)#  
MaxCore(config)#system 1  
MaxCore(system-1)# shelf 1
```

The following table provides the list of shelf mode commands.

Table 5-5 Parameters of Shelf mode commands

| Argument | Description |
|---------------|---|
| BMCIPAddress | IP address of BMC. Set operation is not allowed here; Use Hardware Manager commands. |
| NoOfPCleSlots | Number of PCIe slots available on this shelf. PCIe slots value will be retrieved using BMC API such as getSlotMap |

Table 5-5 Parameters of Shelf mode commands (continued)

| Argument | Description |
|------------------------|---|
| PCleSlot | Configure PCIeSlot |
| ShelfHostIPAddress | IP address of shelf host. Set operation is not allowed here; Use Hardware Manager commands. |
| shelfHostFailurePolicy | Describes the action configured upon the failure of shelf host; Set operation is not allowed here, please use Hardware Manager commands. |
| SATAConfiguration | SATA assignment - CPU address and the corresponding disk Ex: for slot 1 CPU 1 disk 1, the string is Slot 1,CPU 1 - Disk 1; Set operation is not allowed here, please use Hardware Manager commands. |
| USBConfiguration | USB assignment - CPU addresses and the corresponding USBs. Ex: for slot 1 CPU 1 USB 1, the string is Slot 1,CPU 1 - USB 1; Set operation is not allowed here, please use Hardware Manager commands. |
| alarm | Configure alarm. |
| bay | Configure alarm. |
| chassis | Configure chassis. |
| coolingPolicy | Contains the cooling policy for the fans in the chassis. Set operation is not allowed here; Use Hardware Manager commands. |
| inventoryInfo | Contains the shelf inventory information retrieved from the BMC Information will contain manufacturer information, part number, serial number and so on. |
| manufacturerId | Contains the manufacturer information of the shelf. This will be retrieved from shelf RU information using BMC. For example, Artesyn. |
| name | Represents the name of the shelf retrieved from shelf inventory information. For example, MaxCore. |
| NoOfPCleSlots | Number of PCIe slots available on this shelf. PCIe slots value will be retrieved using BMC API such as getSlotMap. |
| partNumber | Contains the part number of the shelf. Value of this Attribute will be retrieved from BMC. For example, AB15467. |
| pexMode | PEX mode - BASE/FABRIC Set operation is not allowed here; Use Hardware Manager commands. |

CLI Commands for PCIeSlot Mode

Table 5-5 Parameters of Shelf mode commands (continued)

| Argument | Description |
|------------------------|--|
| serialNumber | Contains the serial number of the shelf. Value of this Attribute will be retrieved from BMC. For example, 123N45. |
| shelfAddr | Contains the physical location of the shelf. The information can include building number, lab number, rack number and slot number. For example, 1.2.3.5 <building>.<lab>.<rack>.<slot in rack> |
| shelfHostAddress | Contains the shelf host address. Address format is as follows - {slot number <space> cpu number} For example, 1 1 Set operation is not allowed here; Use Hardware Manager commands. |
| shelfHostFailurePolicy | Describes the action configured upon the failure of shelf host. Set operation is not allowed here; Use Hardware Manager commands. |
| shelfPowerStatus | Contains the cold power status of the shelf. Value of this attribute will be retrieved using the BMC API (getFRUInstancePowerState) |
| switchPowerStatus | This method switch on shall move shelf from shelf powered down state to shelf cold start state and vice versa. |
| temperature | This attribute is to show the temperature of the shelf in degrees. For example, 40. |
| version | Contains the product version of the shelf. For example, MaxCore 1.0 and so on. |

5.7 CLI Commands for PCIeSlot Mode

To use the PCIeSlot commands, you need to enter into the PCIeSlot mode. This section describes CLI navigation to enter into PCIeSlot mode.

Syntax

```
MaxCore(config)#  
MaxCore(config)#  
MaxCore(config)#system 1  
MaxCore(system-1)#shelf 1  
MaxCore(shelf-1-1)#PCIeSlot 1
```

The following table provides the list of PCIeSlot mode commands.

Table 5-6 Parameters of PCIeSlot mode commands

| Argument | Description |
|-------------|---|
| PCIeCard | Configure PCIeCard |
| exit | Exit from PCIeSlot configuration mode. |
| IsPopulated | This attribute shows whether slot is populated or not (true/false). |
| name | Represents the name of the slot. For example, slot1.PCIe-7410. |
| no | Negate a command or set its defaults. |
| show | Show running system information. |
| slotMode | This will represent the configured mode (host or endpoint) of the slot. Value will be retrieved by the PEX configuration. |
| slotType | Represents the PCIe specification of the slot information such as version and lanes in conjunction. For example, PCIe3x1 PCIe3x2, PCIe3x4, PCIe3x8, PCIe3x16, PCIe3x32 and so on. |

5.8 CLI Commands for PCIeCard Mode

To use the PCIeCard commands, you need to enter into the PCIeCard mode. This section describes CLI navigation to enter into PCIeCard mode.

Syntax

```
MaxCore(config)#system 1
```

```
MaxCore(system-1)#shelf 1
```

```
MaxCore(shelf-1-1)#PCIeSlot 1
```

```
MaxCore(PCIeSlot-1-1-1)#PCIeCard 1
```

The following table provides the list of PCIeCard mode commands.

Table 5-7 Parameters of PCIeCard mode commands

| Argument | Description |
|-------------------|--|
| CPU | Configure CPU |
| CardConfiguration | Configure Card Configuration. |
| PCIeCardType | Represents the PCIe specification of the PCIe card information such as version and lanes in conjunction. For example, PCIe3x1 PCIe3x2, PCIe3x4, PCIe3x8, PCIe3x16, PCIe3x32 and so on. |

CLI Commands for CPU Mode

Table 5-7 Parameters of PCIeCard mode commands (continued)

| Argument | Description |
|--------------------|---|
| PCleVersion | Contains the version information of the PCIe card. Value of this attribute will be derived from the BMC API. |
| exit | Exit from PCIe Card configuration mode. |
| manufacturerId | Contains the manufacturer information of the PCIe card. This will be derived from the BMC API. For example, Artesyn, Intel and so on. |
| name | Contains the name of the PCIe Card. value of this attribute will be derived from the BMC API. For example, SharpServer, NIC-XL520, NIC-XL710 and so on. |
| no | Negate a command or set its defaults. |
| partNumber | Contains the part number of the PCIe card. Value of this Attribute will be retrieved from BMC API. For example, AB15467 |
| productInformation | Contains the product information of the PCIe card. This will be derived from the BMC API. For example, XL520 (Niantic), XL710 (FortVile). |
| serialNumber | Contains the serial number of the PCIe card. Value of this attribute will be retrieved from the BMC API. |
| show | Show running system information. |

5.9 CLI Commands for CPU Mode

To use the CPU commands, you need to enter into the CPU mode. This section describes CLI navigation to enter into CPU mode.

Syntax

```
MaxCore(config)#system 1
MaxCore(system-1)#shelf 1
MaxCore(shelf-1-1)#PCIEslot 1
MaxCore(PCIEslot-1-1-1)#PCIECard 1
MaxCore(PCIECard-1-1-1-1)#CPU 1
MaxCore(CPU-1-1-1-1-1)#?
```

The following table provides the list of CPU mode commands.

Table 5-8 Parameters of CPU mode commands

| Argument | Description |
|----------|-----------------|
| BIOS | Configure BIOS. |

Table 5-8 Parameters of CPU mode commands (continued)

| Argument | Description |
|-----------------|--|
| CPUInformation | Contains the name of the CPU architecture family and also Operating frequency information. For example, BroadWell 2.10GHz CPU. |
| CPUStatus | Represents the status of the CPU. For example, OK, Failed, HOT and so on. |
| endPoint_hdd_ND | Configure endPoint_hdd_ND. |
| endPoint_nic_ND | Configure endPoint_nic_ND. |
| exit | Exit from CPU configuration mode. |
| isMgmtCapable | Contains the details of the CPU is capable of management. This will be derived from the MCCS isShelfHost. If this CPU is chosen as shelf host by BMC, value will be set to true otherwise false. |
| manufacturerId | Contains the manufacturer details of the CPU. This will be derived from the MCCS. |
| name | Represents the name of the CPU. For example, CPU.1. |
| powerStatus | Power status of CPU - on/off |
| productId | Contains the product details of the CPU. This will be derived from the MCCS. |
| revision | Contains the product information of the CPU. This will be derived from the MCCS. |
| show | Show running system information. |
| virExecEnv | Configure virExecEnv. |
| powerPolicy | Enable or disable power of CPU - On /Off. |

5.10 CLI Commands for BIOS Configuration mode

This section describes CLI navigation to enter into BIOS Configuration mode.

```
#configure terminal
MaxCore(config)#
MaxCore(config)#system 1
MaxCore(system-1)#shelf 1
MaxCore(shelf-1-1)#PCIEslot 1
MaxCore(PCIEslot-1-1-1)#PCIECard 1
MaxCore(PCIECard-1-1-1-1)#CPU 1
MaxCore(CPU-1-1-1-1-1)#BIOS Configuration
```

CLI Commands for BIOS Configuration mode

MaxCore(BIOS-1-1-1-1-1-Configuration)#

The following table provides a brief description about the BIOS mode commands.

Table 5-9 Parameters of BIOS mode commands

| Attribute | Description |
|------------------|--|
| bootOrder | Full name of a boot option, as it is shown by the BIOS. Set operation is not allowed here; Use Hardware Manager commands. |
| baudRate | IOS Setup > Advanced > Console Redirection > Baud Rate Set operation is not allowed here; Use Hardware Manager commands. |
| skipShell | BIOS Setup > Advanced > Boot Configuration > Skip Internal EFI Shell Set operation is not allowed here; Use Hardware Manager commands. |
| autoReset | BIOS Setup > Advanced > Boot Configuration > Auto Reset. Set operation is not allowed here; Use Hardware Manager commands. |
| PCIeSRIOV | BIOS Setup > Advanced > Peripheral Configuration > PCIe SR-IOV Set operation is not allowed here; Use Hardware Manager commands. |
| PCIeARI | BIOS Setup > Advanced > Peripheral Configuration > PCIe ARI Set operation is not allowed here; Use Hardware Manager commands. |
| PCIe64bitDecode | BIOS Setup > Advanced > Peripheral Configuration > Pci 64-bit Decode Set operation is not allowed here; Use Hardware Manager commands. |
| spreadSpectrum | BIOS Setup > Advanced > Peripheral Configuration > Spread Spectrum Set operation is not allowed here; Use Hardware Manager commands. |
| VTD | BIOS Setup > Advanced > SIIO Configuration > Intel VT for Directed I/O (VT-d) > Intel VT for Directed I/O (VT-d) Set operation is not allowed here; Use Hardware Manager commands. |
| interruptMapping | BIOS Setup > Advanced > SIIO Configuration > Intel VT for Directed I/O (VT-d) > Interrupt Remapping Set operation is not allowed here; Use Hardware Manager commands. |
| SATAController | BIOS Setup > Advanced > PCH SATA Configuration > SATA Controller Set operation is not allowed here; Use Hardware Manager commands. |
| SATAMode | BIOS Setup > Advanced > PCH SATA Configuration > HDC Configure As Set operation is not allowed here; Use Hardware Manager commands. |
| RAIDOROMDelay | BIOS Setup > Advanced > PCH SATA Configuration > SATA Mode options > RAID OROM prompt delay Set operation is not allowed here; Use Hardware Manager commands. |
| ALPMSupport | BIOS Setup > Advanced > PCH SATA Configuration > Aggressive LPM Support Set operation is not allowed here; Use Hardware Manager commands. |

CLI Commands for BIOS Configuration mode

Table 5-9 Parameters of BIOS mode commands (continued)

| Attribute | Description |
|----------------------|--|
| SATASpeedSupport | BIOS Setup > Advanced > PCH SATA Configuration > SATA Speed Support Set operation is not allowed here; Use Hardware Manager commands. |
| displayMode | BIOS Setup > Advanced > Video Configuration > Display Mode Set operation is not allowed here; Use Hardware Manager commands. |
| USBSupport | BIOS Setup > Advanced > USB Configuration > USB BIOS Support Set operation is not allowed here; Use Hardware Manager commands. |
| USBPort0 | BIOS Setup > Advanced > USB Configuration > USB Port0 Set operation is not allowed here; Use Hardware Manager commands. |
| USBPort1 | BIOS Setup > Advanced > USB Configuration > USB Port1 Set operation is not allowed here; Use Hardware Manager commands. |
| USBPort2 | BIOS Setup > Advanced > USB Configuration > USB Port2 Set operation is not allowed here; Use Hardware Manager commands. |
| USBPort3 | BIOS Setup > Advanced > USB Configuration > USB Port3 Set operation is not allowed here; Use Hardware Manager commands. |
| USBPort4 | BIOS Setup > Advanced > USB Configuration > USB Port4 Set operation is not allowed here; Use Hardware Manager commands. |
| USBPort5 | BIOS Setup > Advanced > USB Configuration > USB Port5 Set operation is not allowed here; Use Hardware Manager commands. |
| USBPort6 | BIOS Setup > Advanced > USB Configuration > USB Port6 Set operation is not allowed here; Use Hardware Manager commands. |
| USBPort7 | BIOS Setup > Advanced > USB Configuration > USB Port7 Set operation is not allowed here; Use Hardware Manager commands. |
| hyperThreading | BIOS Setup > Advanced > Processor Configuration > Hyper-Threading [ALL] Set operation is not allowed here; Use Hardware Manager commands. |
| processorPerformance | BIOS Setup > Advanced > Processor Configuration > Performance/Watt Set operation is not allowed here; Use Hardware Manager commands. |
| executeDisableBit | Description - BIOS Setup > Advanced > Processor Configuration > Execute Disable Bit Set operation is not allowed here; Use Hardware Manager commands. |
| enableLTSX | BIOS Setup > Advanced > Processor Configuration > Enable LTSX Set operation is not allowed here; Use Hardware Manager commands. |

CLI Commands for BIOS Configuration mode

Table 5-9 Parameters of BIOS mode commands (continued)

| Attribute | Description |
|-----------------------|--|
| VMX | BIOS Setup > Advanced > Processor Configuration > VMX Set operation is not allowed here; Use Hardware Manager commands. |
| hardwarePrefetcher | BIOS Setup > Advanced > Processor Configuration > Hardware Prefetcher Set operation is not allowed here; Use Hardware Manager commands. |
| adjacentCachePrefetch | BIOS Setup > Advanced > Processor Configuration-> Adjacent Cache Prefetch Set operation is not allowed here; Use Hardware Manager commands. |
| directCacheAccess | BIOS Setup > Advanced > Processor Configuration > Direct Cache Access (DCA) Set operation is not allowed here; Use Hardware Manager commands. |
| X2APIC | BIOS Setup > Advanced > Processor Configuration > X2APIC Set operation is not allowed here; Use Hardware Manager commands. |
| EIST | BIOS Setup > Advanced > Advanced Power Management Configuration > EIST (GV3) Set operation is not allowed here; Use Hardware Manager commands. |
| turboMode | BIOS Setup > Advanced > Advanced Power Management Configuration > CPU P State Control > Turbo Mode Set operation is not allowed here; Use Hardware Manager commands. |
| CPUCState | BIOS Setup > Advanced > Advanced Power Management Configuration > CPU C State Control > CPU C State Set operation is not allowed here; Use Hardware Manager commands. |
| CPUCStateLimit | BIOS Setup > Advanced > Advanced Power Management Configuration > CPU C State Control > Package C State limit Set operation is not allowed here; Use Hardware Manager commands. |
| CPUC3Report | BIOS Setup > Advanced > Advanced Power Management Configuration > CPU C State Control > CPU C3 report Set operation is not allowed here; Use Hardware Manager commands. |
| CPUC6Report | BIOS Setup > Advanced > Advanced Power Management Configuration > CPU C State Control > CPU C6 report Set operation is not allowed here; Use Hardware Manager commands. |
| enhancedHaltState | BIOS Setup > Advanced > Advanced Power Management Configuration > CPU C State Control > Enhanced Halt State (C1E) Set operation is not allowed here; Use Hardware Manager commands. |

Table 5-9 Parameters of BIOS mode commands (continued)

| Attribute | Description |
|-------------------|---|
| OSACPICx | BIOS Setup > Advanced > Advanced Power Management Configuration > CPU C State Control > OS ACPI Cx Set operation is not allowed here; Use Hardware Manager commands. |
| DTS | BIOS Setup > Advanced > Thermal Configuration > Cpu Thermal Configuration > DTS Set operation is not allowed here; Use Hardware Manager commands. |
| memoryFrequency | BIOS Setup > Advanced > Memory Configuration-> Memory Frequency Set operation is not allowed here; Use Hardware Manager commands. |
| memoryHalt | BIOS Setup > Advanced > Memory Configuration > Halt on mem Training Error Set operation is not allowed here; Use Hardware Manager commands. |
| numa | BIOS Setup > Advanced > Common RefCode Configuration > Numa Set operation is not allowed here; Use Hardware Manager commands. |
| bootType | Boot > Boot Type Set operation is not allowed here; Use Hardware Manager commands. |
| PXEBoot | Boot > NetworkStack Set operation is not allowed here; Use Hardware Manager commands. |
| PXEBootCapability | Boot > PXE Boot capability Set operation is not allowed here; Use Hardware Manager commands. |
| USBBoot | Boot > USB Boot Set operation is not allowed here; Use Hardware Manager commands. |

5.11 CLI Commands for VEE mode

This section describes CLI navigation to enter into Virtual Execution Environment (VEE) mode.

```
#configure terminal
MaxCore(config)#
MaxCore(config)#system 1
MaxCore(system-1)#shelf 1
MaxCore(shelf-1-1)#PCIEslot 1
MaxCore(PCIEslot-1-1-1)#PCIECard 1
MaxCore(PCIECard-1-1-1-1)#CPU 1
MaxCore(CPU-1-1-1-1-1)#virExecEnv 0
```

CLI Commands for VEE mode

```
MaxCore(virExecEnv-1-1-1-1-1-0)#
```

The following table provides a brief description about the VEE commands.

Table 5-10 Parameters of VEE mode commands

| Attribute | Description |
|-------------------------|---|
| collectLog | Collects the system log files. |
| diagnostic | Configure diagnostic. |
| endPoint_HDD | Configure endPoint_HDD. |
| endPoint_NIC | Configure endPoint_NIC. |
| exit | Exit from virExecEnv configuration mode. |
| firmWareVersion | Firmware version returns the firmware version of the PCIe blade and the application version also value will be retrieved using BBS tool. |
| firmwareUpgrade | Upgrades the Firmware. Parameter will change as per the upgrade procedure provided. |
| getHistoricalSensorData | This functions gets the historical sensor data of the site such as time interval, threshold min, threshold max, current value. |
| getSensorData | This functions gets the sensor data of the site such as time interval, threshold min, threshold max, current value. Sensor data is retrieved using BBS tools. |
| interface | Configure interface. |
| linuxApplication | Configure linuxApplication. |
| listDevices | Lists available devices for firmware upgrade in the blade This will be retrieved using BBS tool. |
| listDisks | Provides a list of disks discovered by the OS along with type, vendor, capacity and usage. |
| listPciEndPoints | This method lists the PCIe end points attached to this VM instance PEX config API will be used for this operation. |
| listSensors | This function returns the list of sensors on PCIe card such as temperature, voltage, power and so on. This list of sensors is retrieved using BBS tools. |
| maxCoreFirmwareUpgrade | Firmware upgrade method upgrades the firmware on boards at each level in the MaxCore hierarchy. |
| maxCoreSystemUpdate | This method accepts the complete update package, unpacks it and performs the updates on a card in sequence. |

Table 5-10 Parameters of VEE mode commands (continued)

| Attribute | Description |
|------------------------|--|
| name | This attribute represents the OS as host OS or vm1, vm2 and so on. |
| no | Negate a command or set its defaults. |
| osVersion | OS Version information of VM running. This will be returned using the Linux command. |
| reboot | To reboot the VEE. This will reboot the VM which results in SSF components restart also. |
| scp | Copies the files from current VM to remote location using Linux scp. |
| show | Show running system information. |
| shutdown | To bring down the current VM. |
| switch | Configure switch. |
| sysInfo | Shows the System Information of the VM such as uptime, date, meminfo and so on. |
| tcpdump | Dump traffic on a network. This method makes use of the linux tcpdump command. |
| updateSensorThresholds | This functions updates the sensor limits of site such as threshold min, threshold max updateSensor threshold values will be done using BBS tool. |

5.12 CLI Commands for Linux application mode

This section describes CLI navigation to enter into Linux Application mode.

```
#configure terminal
MaxCore(config)#
MaxCore(config)#system 1
MaxCore(system-1)#shelf 1
MaxCore(shelf-1-1)#PCIEslot 1
MaxCore(PCIEslot-1-1-1)#PCIECard 1
MaxCore(PCIECard-1-1-1-1)#CPU 1
MaxCore(CPU-1-1-1-1-1)#virExecEnv 0
MaxCore(virExecEnv-1-1-1-1-1-0)#linuxApplication syslcu
MaxCore(linuxApplication-1-1-1-1-1-0-syslcu)#?
```

CLI Commands for Linux application mode

The following table provides a brief description about the Linux Application commands.

Table 5-11 Parameters of Linux Application mode commands

| Attribute | Description |
|-----------------|--|
| commit-config | Commit the configuration for a Linux application to consume. This method also restarts the service with the latest configuration. |
| description | General description about the Linux service describing the functionality of the application. |
| edit-config | Edit the configuration file of a Linux application. |
| exit | Exit from LinuxApplication configuration mode. |
| listConfigFiles | This method returns the list of config files for the Linux application. |
| name | Represents the name of the application. For example, syslog-ng. |
| reload | This method reloads a new configuration of the Linux application without restarting it. |
| restart | This method restarts the Linux application. |
| show | Show running system information. |
| start | This method starts a Linux application. |
| status | Operational Status of the Linux application values are: 0 - STOPPED 1 - RUNNING. |
| stop | This method stops the running Linux application. |
| upload-config | Upload a configuration file to the Core. This is method will not be exposed to user. This will be used for internal development purpose. |

System Log Collection Commands

System Log Collection commands are used to collect system logs from different cards to SSF core and then to download those from SSF core to the user specified locations.

When user triggers `collectlog` command, SSF core will start collecting the logs at the specified location and then it will put the collected log files in `tar.gz` format in SSF core predefined location `/var/log/ssf/system_logs`.

These commands can be executed at any level through out the system hierarchy.

6.1 collectlog

The `collectlog` command collects the required system log files at the specified MOID level and keep the files in tar format at the predefined directory in the SSF core. The predefined directory is `/var/log/ssf/system_logs`. The following are the input arguments required for this command.

Syntax

```
#collectlog filename <filename> size <size>
```

| Argument | Description |
|----------|--|
| filename | The file name with which log tar ball should be created in SSF core location |
| size | The maximum file size to be used for storing the data and it should not exceed the available size in that partition. The available size can be get from 'ssflogsize' command |

6.2 listlog

The `listlog` command lists all the collected log file names with comma (",") separator between each file. It doesn't require any input arguments.

Syntax

```
#listlog
```

6.3 deletelog

The `deletelog` command deletes all the collected log files from the SSF Core. It does not require any input arguments.

Syntax

```
#deletelog
```

6.4 ssflogsize

The `ssflogsize` command provides the free space available in the system partition, where all SSF core collected log files will be placed. The output will be free memory size in MBs. It does not require any input arguments.

Syntax

```
#ssflogsize
```

6.5 downloadlog

The `downloadlog` command downloads the user's selected log file to the specified location using FTP protocol. User has to make sure that the provided remote path should have read-write permissions to copy files to their FTP server location.

Syntax

```
#downloadlog serverip <serverip> username<username> password <password>  
remotepath <path> filename <filename>
```

| Argument | Description |
|----------|---|
| serverip | User server IP address. |
| username | User name for login to the user server. |
| password | Password for the specified user name to login to the user server. |
| path | Location on the user server where the downloaded logs are to be kept. |
| filename | Name of the Log file, which user wants to download from SSF core to user's specified location in path argument. |

Related Documentation

A.1 SMART Embedded Computing Documentation

The documentation listed is referenced in this manual. Technical documentation can be found by using the Documentation Search at <https://www.smartembedded.com/ec/support/> or you can obtain electronic copies of SMART EC documentation by contacting your local sales representative.

Table A-1 SMART Embedded Computing Publications

| Document Title | Publication Number |
|--|---------------------------|
| SSF for MaxCore™ MC3000 Platform XML Interface Guide | 6806800T71 |
| MaxCore™ MC3000 Platform Installation and Use | 6806800T88 |
| MaxCore™ MC3000 Platform Quick Start Guide | 6806800T89 |
| MaxCore™ MC3000 Platform Safety Notes Summary | 6806800T90 |
| Getting Started with MaxCore™ MC3000 Application Note | 6806800T98 |
| MaxCore™ MC3000 Platform Software Installation and Use | 6806800U97 |

