Emulex® Command Line Protocol (CLP)

for XE201, XE501, and XE601 Fibre Channel Controllers

Specification
Revision 11.8

Table of Contents

Chapter 1: Introduction

1.1 Purpose

This document specifies the Broadcom® standard CLP command syntax and operations for Emulex® SLI-4 FC products. These products use the following ASICs: XE201, XE501, and XE601.

1.2 Scope

This document defines the CLP strings and semantics for FC link configurations. Configurations might have a single function per physical link, or in the case of SR-IOV, multiple virtual functions on a physical link.

The Emulex CLP syntax attempts to keep the strings common across the different configurations, and they can be extended when other configurations must be addressed. The syntax is not limited by the number of physical links or by the number of functions present on the device.

Note that CLP strings can be issued to PFs only. It is preferred practice to issue all CLP strings to PF0. CLP strings should not be issued to VFs. The Emulex CLP strings can be specific to the ASIC family, firmware revision, and SKU configuration. As such, the host system might need to be cognizant of the supported behaviors. The host system can use the PCI device identifier to determine which adapter and configuration is present and adjust its behavior accordingly.

1.3 References

- *Distributed Management Task Force (DMTF) Server Management Command Line Protocol (SM CLP) Specification*, Version 1.0.1a Status: Final Status, Publication Date: March 01, 2006, DSP0214
- *PCI Firmware Specification* Revision 3.0, June 20, 2005, available in the online PCI SIG specifications library
- *PCI Option ROM CLP*, final ECN, September 19, 2006, available in the online PCI SIG specifications library

1.4 Abbreviations

BFS	boot from SAN
CLP	command-line protocol
DTMF	Distributed Management Task Force
ECN	engineering change notice
FEC	forward error correction
Gb/s	gigabits per second
NAA	Network Address Authority
PF	physical function
SKU	stock keeping unit
SM	server management
VDM	vendor-defined message
VF	virtual function
XROM	expansion ROM

Chapter 2: CLP Overview

The CLP mechanism is a way to trigger an entry point into the XROM driver. The CLP entry point identifies a PCI bus, device, and function number relative to the command, and it passes two pointers to the XROM driver. The first pointer points to a null-terminated string, which supplies the command to perform and its respective data. The second pointer points to a buffer that can be used for output data storage. The CLP entry point is completed by returning the command execution status data.

The following syntax rules and usage apply to CLP commands:

- A CLP command string typically uses the following format:
  `<verb> <context> <command>[=<value>]`
  where:
  - `<verb>` identifies the type of operation. The following values are valid:
    - `set` – The command is to be set to the `<value>` , which is required in the string.
    - `show` – The command displays the current set value. The `<value>` parameter is not allowed.
    - `exit` – The CLP processing is complete. This verb is required, and it must be the last CLP command issued.
  - `<context>` identifies the target to which the command is addressed, either an entire chip or a physical link. The following values are valid:
    - `/` – The entire chip on the adapter.
    - `fcportx` – A physical link on the adapter.

    For more information, see Section 2.1, Context Identification.
  - `<command>` identifies the specific command that is being set or displayed.
  - `<value>` specifies the new setting for the parameter. It is used only if the `<verb>` is `set`.
- CLP strings are not case-sensitive.

The following conventions apply to CLP output:

- The CLP output data buffer contains a series of ASCII lines followed by at least one null character to make the entire buffer a single null-terminated string. That is, a null character is not contained in or after any line content; it only occurs at the end of the output data.
- Each line is terminated by an ASCII carriage return (`<CR>` value `0x0D`), followed by an ASCII line feed (`<LF>` value `0x0A`).
- The format of the ASCII lines returned is specified by the SM CLP specification.
- Lines are generally formatted relative to the syntax of `<keyword>[=<value>]`.
- The first two lines in the output buffer are usually `status` and `status_tag` keywords that indicate the completion status of the command.
- The `status` and `status_tag` keyword lines are usually followed by command-specific data prefixed by the `retdata` keyword.
- The last line is the `endoutput` keyword.

For example:

```
status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = ELX_2x16G<CR><LF>endoutput<CR><LF><NULL>
```

The Emulex CLP string syntax fully describes the context that is addressed by the command. As such, the function to which the `DMTF_EXEC_CLP_CMD` is issued is largely irrelevant. However, the current design for CLP processing is such that all commands are issued when executing the XROM for function 0 only.

2.1 Context Identification

The CLP syntax specifies a <`context`> for the command. The context is a reference to one of the following items:

- The chip
- An external physical link

The CLP command string identifies the command that is set or queried. CLP command processing validates the command with its respective context (chip or link). If the context is not valid for the command, the CLP command fails.

The physical link references in the CLP strings use the term `fcportx` to indicate the xth external link. This value is in the range 0 to (*n*-1), where *n* is the total number of physical links. The *x* value corresponds to the ordinal position of the physical link as seen on the external faceplate of the adapter. For example, if the faceplate port connectors are viewed as **A** on the top and **B** on the bottom, then port **A** is at ordinal location `0` and port **B** is at ordinal location `1`.

The ordinal position is used because it is not tied to any adapter faceplate labeling scheme, which may vary from model to model. In practice, the ordinal position and the label might be the same (for example, the connector at ordinal position `0` has a label of **0**, and the connector at ordinal position `1` has a label of **1**). However, the distinction is necessary to support other naming conventions that are labelled alphabetically, start link numbering at 1 instead of 0, or are labeled for dual-chip models. For example, in a dual-chip configuration, the faceplate names might be **0** through **3** for the four physical links on the first chip, and **4** through **7** for the four physical links on the second chip. However, the CLP string, which is sent to a specific chip, always uses the range `0` through `3` as its ordinal positions. In the dual-chip example above, the link with faceplate name **4** is referred to as `0` in CLP strings because it is in the first ordinal position on the chip.

2.2 Persistence of Changes

The CLP uses two change persistence catgories:

- Volatile: The setting persists across all resets including chip resets, firmware resets, and function resets, but it does not persist across a power cycle. If the adapter is moved to a different physical system, or if the system is powered off, the setting is lost.
- Nonvolatile: The setting persists across a power cycle and all resets, including chip resets, firmware resets, and function resets. If the adapter is moved to a different physical system the setting continues to exist in the new system.

Chapter 3: CLP Command Reference

This chapter describes the CLP command strings supported by the SLI-4 device. The syntax and semantics of each string are described, and an example is given. A sample response buffer context for each string is provided. CLP processing error values and command status values are obtained from the DMTF SM CLP specification and set appropriately.

CLP command strings are organized by command groups. The following two tables are presented for your convenience. The first table is organized by functional group, and the second table is organized alphabetically by CLP command strings.

Table 1: Command Reference Functional Groups

Command Group	Description	Commands
Device Configuration	These commands could apply to devices other than an FC link, such as an Ethernet NIC (although this document describes Emulex FC HBA behavior only).	Set Default Set OEMELX_AddressDefaults Set OEMELX_LinkConfig Show OEMELX_LinkConfig Set OEMELX_PCIVDM Show OEMELX_PCIVDM Set OEMELX_PortEnable Show OEMELX_PortEnable Show OEMELX_BoardDescription Show OEMELX_FirmwareVersions Show OEMELX_Manufacturer Show OEMELX_ModelName Show OEMELX_SerialNumber Show OEMELX_StandardSpeed

Table 1: Command Reference Functional Groups (Continued)

Command Group	Description	Commands
Fibre Channel Configuration	These commands are specific to an FC link.	Set OEMELX_16GFEC Show OEMELX_16GFEC Set OEMELX_BootEnable Show OEMELX_BootEnable Set OEMELX_LinkSpeed Show OEMELX_LinkSpeed Show OEMELX_MFG_WWNN Show OEMELX_MFG_WWPN Show OEMELX_SpeedCapabilities Set OEMELX_TargetLUN*x* Show OEMELX_TargetLUN*x* Set OEMELX_TargetWWPN*x* Show OEMELX_TargetWWPN*x* Set OEMELX_Topology Show OEMELX_Topology Set OEMELX_VolatileWWNN Show OEMELX_VolatileWWNN Set OEMELX_VolatileWWPN Show OEMELX_VolatileWWPN Set OEMELX_WWNN Show OEMELX_WWNN Set OEMELX_WWPN Show OEMELX_WWPN

Table 2: CLP Command Summary

Verbs	Command String	Description	Context	Persist-ence	Section
`set`	`Default`	Reverts the CLP changeable settings to their factory defaults	`Chip`	NV	Section 3.1.1, set Default
`set`	`OEMELX_16GFEC`	Controls whether FEC is negotiated on a 16Gb/s link	`Physical link`	V	Section 3.2.1, set OEMELX_16GFEC
`show`	`OEMELX_16GFEC`	Reports whether FEC is negotiated on a 16Gb/s link	`Physical link`	V	Section 3.2.2, show OEMELX_16GFEC
`set`	`OEMELX_AddressDefaults`	Resets the WWNs to their manufacturing defaults	`Chip`	NV	Section 3.1.2, set OEMELX_AddressDefaults
`show`	`OEMELX_BoardDescription`	Reports the adapter board description	`Chip`	NV	Section 3.1.3, show OEMELX_BoardDescription
`set`	`OEMELX_BootEnable`	Controls whether FC boot operation is enabled on a link	`Physical link`	V	Section 3.2.3, set OEMELX_BootEnable
`show`	`OEMELX_BootEnable`	Reports whether FC boot operation is enabled on a link	`Physical link`	V	Section 3.2.4, show OEMELX_BootEnable
`show`	`OEMELX_FirmwareVersions`	Reports the firmware version	`Physical link`	N/A	Section 3.1.4, show OEMELX_FirmwareVersions
`set`	`OEMELX_LinkConfig`	Controls the device's link configuration	`Chip`	NV	Section 3.1.5, set OEMELX_LinkConfig

Table 2: CLP Command Summary (Continued)

Verbs	Command String	Description	Context	Persist-ence	Section
show	OEMELX_LinkConfig	Reports the device's link configuration	Chip	NV	Section 3.1.6, show OEMELX_LinkConfig
set	OEMELX_LinkSpeed	Controls the FC link speed to use for boot	Physical link	V	Section 3.2.5, set OEMELX_LinkSpeed
show	OEMELX_LinkSpeed	Reports the FC link speed to use for boot	Physical link	V	Section 3.2.6, show OEMELX_LinkSpeed
show	OEMELX_Manufacturer	Reports the adapter manufacturer	Chip	N/A	Section 3.1.7, show OEMELX_Manufacturer
show	OEMELX_MFG_WWNN	Reports the factory-assigned WWNN value for an FC link	Physical link	N/A	Section 3.2.7, show OEMELX_MFG_WWNN
show	OEMELX_MFG_WWPN	Reports the factory-assigned WWPN value for an FC link	Physical link	N/A	Section 3.2.8, show OEMELX_MFG_WWPN
show	OEMELX_ModelName	Reports the adapter model name	Chip	N/A	Section 3.1.8, show OEMELX_ModelName
set	OEMELX_PCIVDM	Controls the PCIe VDM functionality on a device	Chip	NV	Section 3.1.9, set OEMELX_PCIVDM
show	OEMELX_PCIVDM	Reports the PCIe VDM functionality on a device	Chip	NV	Section 3.1.10, show OEMELX_PCIVDM
set	OEMELX_PortEnable	Controls physical link enablement	Physical link	NV	Section 3.1.11, set OEMELX_PortEnable
show	OEMELX_PortEnable	Reports physical link enablement	Physical link	NV	Section 3.1.12, show OEMELX_PortEnable
show	OEMELX_SerialNumber	Reports the adapter serial number	Chip	N/A	Section 3.1.13, show OEMELX_SerialNumber
show	OEMELX_SpeedCapabilities	Reports the FC link speeds supported by the adapter and installed SFP	Physical link	N/A	Section 3.2.9, show OEMELX_SpeedCapabilities
show	OEMELX_StandardSpeed	Reports the standard speed for the FC chip	Chip	N/A	Section 3.1.14, show OEMELX_StandardSpeed
set	OEMELX_TargetLUNx	Controls the SCSI LUN for boot target *x* for an FC link.	Physical link	V	Section 3.2.10, set OEMELX_TargetLUN[x]
show	OEMELX_TargetLUNx	Reports the SCSI LUN for boot target *x* for an FC link.	Physical link	V	Section 3.2.11, show OEMELX_TargetLUN[x]
set	OEMELX_TargetWWPNx	Controls the WWPN for boot target *x* for an FC link	Physical link	V	Section 3.2.12, set OEMELX_TargetWWPN[x]
show	OEMELX_TargetWWPNx	Reports the WWPN for boot target *x* for an FC link	Physical link	V	Section 3.2.13, show OEMELX_TargetWWPN[x]
set	OEMELX_Topology	Controls the FC link topology to use for boot	Physical link	V	Section 3.2.14, set OEMELX_Topology
show	OEMELX_Topology	Reports the FC link topology to use for boot	Physical link	V	Section 3.2.15, show OEMELX_Topology
set	OEMELX_VolatileWWNN	Controls the volatile WWNN value for an FC link	Physical link	V	Section 3.2.16, set OEMELX_VolatileWWNN
show	OEMELX_VolatileWWNN	Reports the volatile WWNN value for an FC link	Physical link	V	Section 3.2.17, show OEMELX_VolatileWWNN
set	OEMELX_VolatileWWPN	Controls the volatile WWPN value for an FC link	Physical link	V	Section 3.2.18, set OEMELX_VolatileWWPN

Table 2: CLP Command Summary (Continued)

Verbs	Command String	Description	Context	Persist-ence	Section
`show`	`OEMELX_VolatileWWPN`	Reports the volatile WWPN value for an FC link	`Physical link`	V	Section 3.2.19, show OEMELX_VolatileWWPN
`set`	`OEMELX_WWNN`	Sets the nonvolatile WWNN value for an FC link	`Physical link`	NV	Section 3.2.20, set OEMELX_WWNN
`show`	`OEMELX_WWNN`	Reports the current volatile, nonvolatile, or factory WWNN for an FC link	`Physical link`	NV	Section 3.2.21, show OEMELX_WWNN
`set`	`OEMELX_WWPN`	Sets the nonvolatile WWPN value for an FC link	`Physical link`	NV	Section 3.2.22, set OEMELX_WWPN
`show`	`OEMELX_WWPN`	Reports the current volatile, nonvolatile, or factory WWPN for an FC link	`Physical link`	NV	Section 3.2.23, show OEMELX_WWPN

NOTE: WWNNs and WWPNs are specified in network byte order. The first byte in the string contains the NAA field, and so on. For example, `10000000C9D1B04B`.

NOTE: The `show` functionality for a command that uses a volatile setting fails if the corresponding `set` command has not yet been run.

3.1 Device Configuration Commands

3.1.1 set Default

This command resets all of the chip parameters modified by the CLP to their factory-assigned defaults.

This command clears only the parameters that have been set by CLP commands. If a host had modified any parameters outside of the CLP (for example, through a BIOS configuration utility or an operating-system tool), those items might become "owned" by the respective tool chain and thus might not be reset by this command. Only the values set exclusively by the CLP commands are sure to be cleared by this command.

Syntax

Input:

`set <context> Default`

NOTE: Because there is no returned data, no command-specific `retdata = <value>` field is returned. Only the command completion status is returned.

Persistance

Nonvolatile

Parameters

`context` Chip = /

Examples

Input string:
```
set / Defaults
```

Output buffer:
```
status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>endoutput<CR><LF><NULL>
```

3.1.2 set OEMELX_AddressDefaults

This command sets the WWNNs and WWPNs for all PFs on the device to their manufacturing defaults.

Syntax

Input:
```
set <context> OEMELX_AddressDefaults
```

Output:
```
retdata = 0000000000000000
```

Persistance

Nonvolatile

Parameters

`context` Chip = /

Examples

Input string:
```
set / OEMELX_AddressDefaults
```

Output buffer:
```
status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = 0000000000000000<CR><LF>endoutput<CR><LF><NULL>
```

3.1.3 show OEMELX_BoardDescription

This command reports the board description.

Syntax

Input:
```
show <context> OEMELX_BoardDescription
```

Output:
```
retdata = <value>
```

Persistance

N/A

Parameters

`context`	Chip = `/`
`value`	ASCII string containing the board description, which can contain spaces.

Examples

Input string:

```
show / OEMELX_BoardDescription
```

Output buffer:

```
status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = Emulex LPe35002-M2 2-Port 32Gb Fibre
Channel Adapter<CR><LF>endoutput<CR><LF><NULL>
```

3.1.4 show OEMELX_FirmwareVersions

This command reports the version of the firmware on the device.

Syntax

Input:

```
show <context> OEMELX_FirmwareVersions
```

Output:

```
retdata = <value>
```

Persistance

N/A

Parameters

`context`	Physical link *x* = `fcport`*x*
`value`	Firmware version string.

Examples

Input string:

```
show fcport0 OEMELX_FirmwareVersions
```

Output buffer:

```
status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = 12.8.351.44<CR><LF>endoutput<CR><LF><NULL>
```

3.1.5 set OEMELX_LinkConfig

This command sets the value of the device's link configuration. It affects chip-global configuration changes. This command is valid only on an SKU that supports reconfigurable links, such as a link that can be converted between a 2x64GFC interface and a 4x32GFC interface.

If the command changes the device's configuration, the new setting takes effect at the next power cycle, PCI hot reset, or PCI fundamental reset. Additionally, all prior nonvolatile and volatile WWNN and WWPN values are deleted, and no further settings can be changed until the device has been reset and activates the link configuration change.

Syntax

Input:
`set <context> OEMELX_LinkConfig=<value>`

Output:
`retdata = <value>`

Persistance

Nonvolatile

Parameters

`context`	Chip = `/`
`value`	2x64GFC = `ELX_2x64G` 2x32GFC = `ELX_2x32G` 2x16GFC = `ELX_2x16G` 4x32GFC = `ELX_4x32G` 4x16GFC = `ELX_4x16G` 4x8G FC = `ELX_4x8G`

Examples

Input string:
`set / OEMELX_LinkConfig=ELX_2x64G`

Output buffer:
`status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = ELX_2x64G<CR><LF>endoutput<CR><LF><NULL>`

3.1.6 show OEMELX_LinkConfig

The command queries the value of the device's link configuration. This parameter affects chip-global configuration changes. This command is valid only on an SKU that supports reconfigurable links, such as a link that can convert between a 2x64GFC interface and a 4x32GFC interface.

Syntax

Input:
`show <context> OEMELX_LinkConfig`

Output:
`retdata = <value>`

Persistance

N/A

Parameters

`context`	Chip = `/`
`value`	2x64GFC = `ELX_2x64G` 2x32GFC = `ELX_2x32G` 2x16GFC = `ELX_2x16G` 4x32GFC = `ELX_4x32G` 4x16GFC = `ELX_4x16G` 4x8G FC = `ELX_4x8G`

Examples

Input string:

```
show / OEMELX_LinkConfig
```

Output buffer:

```
status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = ELX_2x64G<CR><LF>endoutput<CR><LF><NULL>
```

3.1.7 show OEMELX_Manufacturer

This command reports the device manufacturer.

Syntax

Input:

```
show <context> OEMELX_Manufacturer
```

Output:

```
retdata = <value>
```

Persistance

N/A

Parameters

`context`	Chip = `/`
`value`	ASCII string containing the adapter manufacturer

Examples

Input string:

```
show / OEMELX_Manufacturer
```

Output buffer:

```
status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = Broadcom<CR><LF>endoutput<CR><LF><NULL>
```

3.1.8 show OEMELX_ModelName

This command reports the device model name.

Syntax

Input:
`show <context> OEMELX_ModelName`

Output:
`retdata = <value>`

Persistance

N/A

Parameters

`context`	Chip = `/`
`value`	ASCII string containing the model name.

Examples

Input string:
`show / OEMELX_ModelName`

Output buffer:
`status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = LPe35002-M2<CR><LF>endoutput<CR><LF><NULL>`

3.1.9 set OEMELX_PCIVDM

This command configures the PCIe VDM capability for the device. If the command changes the device's configuration, the new setting takes effect at the next power-cycle, PCI hot reset, or PCI fundamental reset. The setting is saved in nonvolatile memory and persists across all power cycles of the board.

If PCIe VDM is enabled and the board is moved to a system that does not support PCIe VDM, the system might have issues with the board. PCIe VDM must be disabled before moving the board to a system that does not support PCIe VDM.

Syntax

Input:
`set <context> OEMELX_PCIVDM=<value>`

Output:
`retdata = <value>`

Persistance

Nonvolatile

Parameters

`context`	Chip = /
`value`	PCIe VDM is enabled on the device = `enabled` PCIe VDM is disabled on the device = `disabled`

Examples

Input string:

```
set / OEMELX_PCIVDM=disabled
```

Output buffer:

```
status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = disabled<CR><LF>endoutput<CR><LF><NULL>
```

3.1.10 show OEMELX_PCIVDM

The command reports the PCIe VDM configuration for the device.

Syntax

Input:

```
show <context> OEMELX_PCIVDM
```

Output:

```
retdata = <value>
```

Persistance

N/A

Parameters

`context`	Chip = /
`value`	PCIe VDM is enabled on the device = `enabled` PCIe VDM is disabled on the device = `disabled`

Examples

Input string:

```
show / OEMELX_PCIVDM
```

Output buffer:

```
status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = disabled<CR><LF>endoutput<CR><LF><NULL>
```

3.1.11 set OEMELX_PortEnable

This command sets the port enablement configuration for the indicated physical link. If the physical link is disabled, the link remains down with the transmitter off.

Syntax

Input:

```
set <context> OEMELX_PortEnable=<value>
```

Output:
`retdata = <value>`

Persistance

Nonvolatile

Parameters

`context`	Physical link *x* = `fcportx`
`value`	Operation is enabled on the corresponding physical link = `enabled` Operation is disabled on the corresponding physical link = `disabled`

Examples

Input string:
`set fcport0 OEMELX_PortEnable=enabled`

Output buffer:
`status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = enabled<CR><LF>endoutput<CR><LF><NULL>`

3.1.12 show OEMELX_PortEnable

This command reports the port enablement configuration for the indicated physical link.

Syntax

Input:
`show <context> OEMELX_PortEnable`

Output:
`retdata = <value>`

Persistance

N/A

Parameters

`context`	Physical link *x* = `fcportx`
`value`	Operation is enabled on the corresponding physical link = `enabled` Operation is disabled on the corresponding physical link = `disabled`

Examples

Input string:
`show fcport0 OEMELX_PortEnable`

Output buffer:
`status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = enabled<CR><LF>endoutput<CR><LF><NULL>`

NOTE: For compatibility with older firmware, the host software should match `enable` or `enabled` with `disable` or `disabled`.

3.1.13 show OEMELX_SerialNumber

This command reports the device serial number.

Syntax

Input:
`show <context> OEMELX_SerialNumber`

Output:
`retdata = <value>`

Persistance

N/A

Parameters

`context`	Chip = `/`
`value`	ASCII string containing the adapter serial number.

Examples

Input string:
`show / OEMELX_SerialNumber`

Output buffer:
`status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = RB50018376<CR><LF>endoutput<CR><LF><NULL>`

3.1.14 show OEMELX_StandardSpeed

This command reports the standard FC link speed for the chip.

Syntax

Input:
`set <context> OEMELX_StandardSpeed`

Output:
`retdata = <value>`

Persistance

N/A

Parameters

`context` chip = /

`value` A 4-digit hexadecimal ASCII string corresponding to the standard link speed for the chip. The following values are valid:
LPe15000 series 8G adapters (XE201) = `0008`
LPe16000 series 16G adapters (XE201) = `0010`
LPe31000 series 16G adapters (XE501) = `0010`
LPe32000 series 32G adapters (XE501) = `0020`
LPe35000 series 32G adapters (XE601) = `0020`
LPe36000 series 64G adapters (XE601) = `0040`

Examples

Input string:

```
set / OEMELX_StandardSpeed
```

Output buffer:

```
status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = 0040<CR><LF>endoutput<CR><LF><NULL>
```

3.2 Fibre Channel Configuration Commands

3.2.1 set OEMELX_16GFEC

This command sets the 16G FEC mode for the indicated physical link. If 16G FEC is disabled, FEC is not used when the link is running at a 16Gb/s link speed.

If 16G FEC is enabled, FEC support is negotiated with the link peer device when the link is running at a 16Gb/s link speed. If the link peer device supports 16G FEC, FEC is used when the link runs at a 16Gb/s link speed. If the link peer device does not support 16G FEC, FEC is not used when the link runs at a 16Gb/s link speed.

The 16G FEC mode value is saved in a volatile store that persists across all resets other than a power cycle. For example, the value persists across a PCI hot reset, PCI fundamental reset (PERST), firmware reset, or function reset.

Syntax

Input:

```
set <context> OEMELX_16GFEC=<value>
```

Output:

```
retdata = <value>
```

Persistance

Volatile

Parameters

`context` Physical link *x* = `fcportx`

`value` 16G FEC is enabled on the device = `enabled`
16G FEC is disabled on the device = `disabled`

Examples

Input string:
`set fcport0 OEMELX_16GFEC=enabled`

Output buffer:
`status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = enabled<CR><LF>endoutput<CR><LF><NULL>`

3.2.2 show OEMELX_16GFEC

This command reports the 16G FEC mode setting for the indicated physical link.

Syntax

Input:
`show <context> OEMELX_16GFEC`

Output:
`retdata = <value>`

Persistance

N/A

Parameters

`context`	Physical link *x* = `fcportx`
`value`	16G FEC is enabled on the device = `enabled` 16G FEC is disabled on the device = `disabled`

Examples

Input string:
`show fcport0 OEMELX_16GFEC`

Output buffer:
`status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = enabled<CR><LF>endoutput<CR><LF><NULL>`

3.2.3 set OEMELX_BootEnable

This command sets the FC boot enablement mode for the indicated physical link. This setting is queried by the boot driver, and it controls whether the physical link is used for boot from SAN. If this setting is enabled, an FC boot target and LUN must also be configured on the physical link.

Syntax

Input:
`set <context> OEMELX_BootEnable=<value>`

Output:
`retdata = <value>`

Persistance

Volatile

Parameters

`context`	Physical link *x* = `fcportx`
`value`	FC boot operation is enabled on the device = `enabled` FC boot operation is disabled on the device = `disabled`

Examples

Input string:

```
set fcport0 OEMELX_BootEnable=enabled
```

Output buffer:

```
status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = enabled<CR><LF>endoutput<CR><LF><NULL>
```

3.2.4 show OEMELX_BootEnable

This command reports the FC boot enablement mode setting for the indicated physical link.

Syntax

Input:

```
show <context> OEMELX_BootEnable
```

Output:

```
retdata = <value>
```

Persistance

N/A

Parameters

`context`	Physical link *x* = `fcportx`
`value`	FC boot operation is enabled on the device = `enabled` FC boot operation is disabled on the device = `disabled`

Examples

Input string:

```
show fcport0 OEMELX_BootEnable
```

Output buffer:

```
status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = enabled<CR><LF>endoutput<CR><LF><NULL>
```

3.2.5 set OEMELX_LinkSpeed

This command sets the FC boot link speed to be used on the physical link. The link speed information is saved in a volatile store that persists across all resets other than a power cycle. For example, the value persists over a PCI hot reset, PCI fundamental reset (PERST), firmware reset, or function reset. By default, the link speed is automatically negotiated to run at the highest speed supported by all devices on the link.

Syntax

Input:
`set <context> OEMELX_LinkSpeed=<value>`

Output:
`retdata = <value>`

Persistance

Volatile

Parameters

`context` Physical link *x* = `fcportx`

`value` A 4-digit hexadecimal string corresponding to the FC boot link speed. The following values are valid:
Auto-negotiate: `0000`
1 Gb/s: `0001`
2 Gb/s: `0002`
4 Gb/s: `0004`
8 Gb/s: `0008`
16 Gb/s: `0010`
32 Gb/s: `0020`
64 Gb/s: `0040`
The speed value must be either `0000` (auto-negotiate) or one of the speed values returned by the `OEMELX_SpeedCapabilities` command.

Examples

Input string:
`set fcport0 OEMELX_LinkSpeed=0020`

Output buffer:
`status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = 0020<CR><LF>endoutput<CR><LF><NULL>`

3.2.6 show OEMELX_LinkSpeed

This command reports the FC boot link speed used on the physical link.

NOTE: If the adapter is configured for trunking, the speed value reflects the link speed to which each physical link within the trunk is set.

Syntax

Input:
`show <context> OEMELX_LinkSpeed`

Output:
`retdata = <value>`

Persistance

N/A

Parameters

`context`	Physical link *x* = `fcportx`
`value`	A 4-digit hexadecimal string corresponding to the FC boot link speed. Refer to Section 3.2.5, set OEMELX_LinkSpeed, for valid values.

Examples

Input string:
`show fcport0 OEMELX_LinkSpeed`

Output buffer:
`status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = 0020<CR><LF>endoutput<CR><LF><NULL>`

3.2.7 show OEMELX_MFG_WWNN

This command reports the Emulex permanent, factory-assigned WWNN for the indicated physical link. This might or might not match the currently active WWNN.

Syntax

Input:
`show <context> OEMELX_MFG_WWNN`

Output:
`retdata = <value>`

Persistance

N/A

Parameters

`context`	Physical link *x* = `fcportx`
`value`	A 16-digit hexadecimal string corresponding to the FC WWNN.

Examples

Input string:
`show fcport0 OEMELX_MFG_WWNN`

Output buffer:
`status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = 20000090FA0DB047<CR><LF>endoutput<CR><LF><NULL>`

3.2.8 show OEMELX_MFG_WWPN

This command reports the Emulex permanent, factory-installed WWPN for the indicated physical link. This might or might not match the currently active WWPN.

Syntax

Input:
`show <context> OEMELX_MFG_WWPN`

Output:
`retdata = <value>`

Persistance

N/A

Parameters

`context`	Physical link *x* = `fcportx`
`value`	A 16-digit hexadecimal string corresponding to the FC WWPN.

Examples

Input string:
`show fcport0 OEMELX_MFG_WWPN`

Output buffer:
`status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = 10000090FA0DB047<CR><LF>endoutput<CR><LF><NULL>`

3.2.9 show OEMELX_SpeedCapabilities

This command reports the physical link speeds supported by the adapter and the port's SFP.

Syntax

Input:
`show <context> OEMELX_SpeedCapabilities`

Output:
`retdata = <value>`

Persistance

N/A

Parameters

`context` Physical link *x* = `fcportx`

`value` A string reporting GC speed values. The following specifications apply:

- The string contains one, two, or three FC speed values.
- If more than one speed value is returned, a `/` character is used between values.
- Multiple speeds are ordered such that the highest speed value is reported first, followed by the next-highest speed value, followed by the lowest speed value.
- If no SFP is installed, the string is set to `noSFP`.
- If the speed capabilities of the physical link cannot be determined, the string is set to `unknown`.
- If the adapter is configured for trunking, the speed values reflect a single physical link within the trunk, and they contain only the values that are common to all physical links in the trunk.

Examples

Input string:

```
show fcport0 OEMELX_SpeedCapabilities
```

Output buffer:

```
status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = 32/16/8<CR><LF>endoutput<CR><LF><NULL>
status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = 16<CR><LF>endoutput<CR><LF><NULL>
status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = 32/16<CR><LF>endoutput<CR><LF><NULL>
status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = noSFP<CR><LF>endoutput<CR><LF><NULL>
status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = unknown<CR><LF>endoutput<CR><LF><NULL>
```

3.2.10 set OEMELX_TargetLUN[x]

This command registers the SCSI LUN for a Fibre Channel boot target to be used for BFS. The boot information is saved in a volatile store that persists across all resets other than a power cycle. For example, the value persists over a PCI hot reset, PCI fundamental reset (PERST), firmware reset, or function reset.

Up to eight boot LUNs, which are accessed in sequential order, can be assigned. You must register an FC target for each boot LUN using the `OEMELX_TargetWWPN[x]` command. See Section 3.2.12, set OEMELX_TargetWWPN[x] for more information.

Syntax

Input:

```
set <context> OEMELX_TargetLUN[x]=<value>
```

Output:

```
retdata = <value>
```

Persistance

Volatile

Parameters

`context` Physical link *x* = `fcportx`

`[x]` First boot LUN = `[0]`
Second boot LUN = `[1]`
Third boot LUN = `[2]`
Fourth boot LUN = `[3]`
Fifth boot LUN = `[4]`
Sixth boot LUN = `[5]`
Seventh boot LUN = `[6]`
Eighth boot LUN = `[7]`

`value` A 4-digit hexadecimal string corresponding to the bootable LUN for the selected boot target.
The following specifications apply:

- Unused digits are padded with zeroes.
- LUNs between `0000` and `FFFF` are supported.
- The string is specified in network byte order.
- The LUN is converted to an 8-byte SCSI LUN structure by using the provided value as the upper 16 bits and using zeroes in the lower 48 bits. For example:
  - A target LUN value of `000F` results in a SCSI LUN value of `00 0F 00 00 00 00 00 00`.
  - A target LUN value of `100F` results in a SCSI LUN value of `10 0F 00 00 00 00 00 00`.

Examples

Input string:

```
set fcport0 OEMELX_TargetLUN[0]=000F
```

Output buffer:

```
status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = 000F<CR><LF>endoutput<CR><LF><NULL>
```

3.2.11 show OEMELX_TargetLUN[x]

This command reports the LUN for the indicated FC boot target. Up to eight boot targets can be reported.

Syntax

Input:

```
show <context> OEMELX_TargetLUN[x]
```

Output:

```
retdata = <value>
```

Persistance

N/A

Parameters

`context`	Physical link *x* = `fcportx`
`[x]`	First boot LUN = `[0]` Second boot LUN = `[1]` Third boot LUN = `[2]` Fourth boot LUN = `[3]` Fifth boot LUN = `[4]` Sixth boot LUN = `[5]` Seventh boot LUN = `[6]` Eighth boot LUN = `[7]`
`value`	A 4-digit hexadecimal string corresponding to the bootable LUN for the selected boot target. For more information, see Section 3.2.10, set OEMELX_TargetLUN[x].

Examples

Input string:

```
show fcport0 OEMELX_TargetLUN[0]
```

Output buffer:

```
status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = 000F<CR><LF>endoutput<CR><LF><NULL>
```

3.2.12 set OEMELX_TargetWWPN[x]

This command registers the WWPN for a Fibre Channel boot target to be used for BFS. The boot information is saved in a volatile store that persists across all resets other than a power cycle. For example, the value persists over a PCI hot reset, PCI fundamental reset (PERST), firmware reset, or function reset.

Up to eight boot targets, which are accessed in sequential order, can be assigned. You must register a LUN for each boot target using the `OEMELX_TargetLUN[x]` command. See Section 3.2.10, set OEMELX_TargetLUN[x], for more information.

Syntax

Input:

```
set <context> OEMELX_TargetWWPN[x]=<value>
```

Output:

```
retdata = <value>
```

Persistance

Volatile

Parameters

`context`	Physical link *x* = `fcportx`
`[x]`	First boot target = `[0]` Second boot target = `[1]` Third boot target = `[2]` Fourth boot target = `[3]` Fifth boot target = `[4]` Sixth boot target = `[5]` Seventh boot target = `[6]` Eighth boot target = `[7]`
`value`	A 16-digit hexadecimal string corresponding to the target FC WWPN.

Examples

Input string:

```
set fcport0 OEMELX_TargetWWPN[0]=10000090FA0DB047
```

Output buffer:

```
status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = 10000090FA0DB047<CR><LF>endoutput<CR><LF><NULL>
```

3.2.13 show OEMELX_TargetWWPN[x]

This command reports the WWPN for the indicated FC boot target. Up to eight boot targets can be reported.

Syntax

Input:

```
show <context> OEMELX_TargetWWPN[x]
```

Output:

```
retdata = <value>
```

Persistance

N/A

Parameters

`context`	Physical link *x* = `fcportx`
`[x]`	First boot target = `[0]` Second boot target = `[1]` Third boot target = `[2]` Fourth boot target = `[3]` Fifth boot target = `[4]` Sixth boot target = `[5]` Seventh boot target = `[6]` Eighth boot target = `[7]`
`value`	A 16-digit hexadecimal string corresponding to the target FC WWPN.

Examples

Input string:

```
show fcport0 OEMELX_TargetWWPN[0]
```

Output buffer:

```
status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = 10000090FA0DB047<CR><LF>endoutput<CR><LF><NULL>
```

3.2.14 set OEMELX_Topology

This command sets the boot link topology to be used for the physical link. The topology information is saved in a volatile store that persists across all resets other than a power cycle. For example, the value persists over a PCI hot reset, PCI fundamental reset (PERST), firmware reset, or function reset. By default, the boot link topology is automatically selected.

NOTE: This command has no efffect on adapters that use the XE501 or XE601 ASIC.

Syntax

Input:

```
set <context> OEMELX_Topology=<value>
```

Output:

```
retdata = <value>
```

Persistance

Volatile

Parameters

`context`	Physical link *x* = `fcportx`
`value`	A 2-digit hexadecimal string corresponding to the desired topology. The following values are valid: Auto-topology, with loop first: `00` Point-to-point: `04` Auto-topology, with point-to-point first: `08` FC-AL: `0C`

Examples

Input string:

```
set fcport0 OEMELX_Topology=08
```

Output buffer:

```
status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = 08<CR><LF>endoutput<CR><LF><NULL>
```

3.2.15 show OEMELX_Topology

This command reports the boot link topology used on the physical link.

Syntax

Input:
`show <context> OEMELX_Topology`

Output:
`retdata = <value>`

Persistance

N/A

Parameters

`context`	Physical link *x* = `fcportx`
`value`	A 2-digit hexadecimal string corresponding to the boot link topology. Refer to Section 3.2.14, set OEMELX_Topology, for valid values.

Examples

Input string:
`show fcport0 OEMELX_Topology`

Output buffer:
`status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = 08<CR><LF>endoutput<CR><LF><NULL>`

3.2.16 set OEMELX_VolatileWWNN

This command sets the volatile FC WWNN for the indicated physical link.

The volatile WWNN value supersedes any nonvolatile WWNN setting, including the factory-default WWNN assigned to the physical link. After it is set, the volatile WWNN value persists across all resets other than a power cycle. For example, the volatile WWNN persists across a PCI hot reset, PCI fundamental reset (PERST), firmware reset, or function reset.

Syntax

Input:
`set <context> OEMELX_VolatileWWNN=<value>`

Output:
`retdata = <value>`

Persistance

Volatile

Parameters

`context`	Physical link *x* = `fcportx`
`value`	A 16-digit hexadecimal string corresponding to the FC WWNN.

Examples

Input string:
`set fcport0 OEMELX_VolatileWWNN=20000090FA0DB047`

Output buffer:
`status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = 20000090FA0DB047<CR><LF>endoutput<CR><LF><NULL>`

3.2.17 show OEMELX_VolatileWWNN

This command reports the volatile FC WWNN for the indicated physical link. If a volatile WWNN has not been set, an error is returned. See Chapter 4 for information about error codes.

Syntax

Input:
`show <context> OEMELX_VolatileWWNN`

Output:
`retdata = <value>`

Persistance

N/A

Parameters

`context`	Physical link *x* = `fcportx`
`value`	A 16-digit hexadecimal string corresponding to the FC WWNN.

Examples

Input string:
`show fcport0 OEMELX_VolatileWWNN`

Output buffer:
`status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = 20000090FA0DB047<CR><LF>endoutput<CR><LF><NULL>`

3.2.18 set OEMELX_VolatileWWPN

This command sets the volatile FC WWPN for the indicated physical link.

The volatile WWPN value supersedes any nonvolatile WWPN setting, including the factory-default WWPN assigned to the physical link. After it is set, the volatile WWPN value persists across all resets other than a power cycle. For example, the volatile WWPN persists across a PCI hot reset, PCI fundamental reset (PERST), firmware reset, or function reset.

Syntax

Input:
`set <context> OEMELX_VolatileWWPN=<value>`

Output:
`retdata = <value>`

Persistance

Volatile

Parameters

`context`	Physical link *x* = `fcportx`
`value`	A 16-digit hexadecimal string corresponding to the FC WWPN.

Examples

Input string:

```
set fcport0 OEMELX_VolatileWWPN=10000090FA0DB047
```

Output buffer:

```
status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = 10000090FA0DB047<CR><LF>endoutput<CR><LF><NULL>
```

3.2.19 show OEMELX_VolatileWWPN

This command reports the volatile FC WWPN for the indicated physical link. If a volatile WWPN has not been set, an error is returned. See Chapter 4 for information about error codes.

Syntax

Input:

```
show <context> OEMELX_VolatileWWPN
```

Output:

```
retdata = <value>
```

Persistance

N/A

Parameters

`context`	Physical link *x* = `fcportx`
`value`	A 16-digit hexadecimal string corresponding to the FC WWPN.

Examples

Input string:

```
show fcport0 OEMELX_VolatileWWPN
```

Output buffer:

```
status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = 10000090FA0DB047<CR><LF>endoutput<CR><LF><NULL>
```

3.2.20 set OEMELX_WWNN

This command sets the nonvolatile FC WWNN for the indicated physical link.

NOTE: The factory-assigned WWNN is not overwritten. It remains in a separate, protected area.—

The nonvolatile WWNN value supersedes the factory-default WWNN assigned to the physical link. After it is set, the nonvolatile WWNN value persists across all resets, including a power cycle. Thus, if the adapter is moved to a different system, the set value persists.

Syntax

Input:
`set <context> OEMELX_WWNN=<value>`

Output:
`retdata = <value>`

Persistance

Nonvolatile

Parameters

`context`	Physical link *x* = `fcportx`
`value`	A 16-digit hexadecimal string corresponding to the FC WWNN.

Examples

Input string:
`set fcport1 OEMELX_WWNN=20000090FA0DB047`

Output buffer:
`status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = 20000090FA0DB047<CR><LF>endoutput<CR><LF><NULL>`

3.2.21 show OEMELX_WWNN

This command reports the current FC WWNN for the indicated physical link. The command does not identify the method used to define the WWNN. For example, it could be the factory-default WWNN, the nonvolatile WWNN, or the volatile WWNN, depending on which is currently set.

Syntax

Input:
`show <context> OEMELX_WWNN`

Output:
`retdata = <value>`

Persistance

N/A

Parameters

`context`	Physical link *x* = `fcportx`
`value`	A 16-digit hexadecimal string corresponding to the FC WWNN.

Examples

Input string:

```
show fcport0 OEMELX_WWNN
```

Output buffer:

```
status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = 20000090FA0DB047<CR><LF>endoutput<CR><LF><NULL>
```

3.2.22 set OEMELX_WWPN

This command sets the nonvolatile FC WWPN for the indicated physical link.

NOTE: The factory-assigned WWPN is not overwritten. It remains in a separate, protected area.

The nonvolatile WWPN value supersedes the factory-default WWPN assigned to the physical link. After it is set, the nonvolatile WWPN value persists across all resets, including a power cycle. Thus, if the adapter is moved to a different system, the set value persists.

Syntax

Input:

```
set <context> OEMELX_WWPN=<value>
```

Output:

```
retdata = <value>
```

Persistance

Nonvolatile

Parameters

`context`	Physical link *x* = `fcportx`
`value`	A 16-digit hexadecimal string corresponding to the FC WWPN.

Examples

Input string:

```
set fcport0 OEMELX_WWPN=10000090FA0DB047
```

Output buffer:

```
status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = 10000090FA0DB047<CR><LF>endoutput<CR><LF><NULL>
```

3.2.23 show OEMELX_WWPN

This command reports the current FC WWPN for the indicated physical link. The command does not identify the method used to define the WWPN. For example, it could be the factory=default WWPN, the nonvolatile WWPN, or the volatile WWPN, depending on which is currently set.

Syntax

Input:

`show <context> OEMELX_WWPN`

Output:

`retdata = <value>`

Persistance

N/A

Parameters

`context`	Physical link *x* = `fcportx`
`value`	A 16-digit hexadecimal string corresponding to the FC WWPN.

Examples

Input string:

```
show fcport0 OEMELX_WWPN
```

Output buffer:

```
status=0<CR><LF>status_tag=COMMAND_COMPLETED<CR><LF>retdata = 10000090FA0DB047<CR><LF>endoutput<CR><LF><NULL>
```

Chapter 4: CLP Completion Codes

This section describes CLP completion codes, as indicated by the ECN on the PCI-SIG website. Emulex adapters, which use a PCI 3.0-compliant CLP implementation for control, return good and error status codes as indicated in Table 3.

The following codes, which are used in Table 3, are specified by the DTMF SM CLP specification:

- a1 = `0`: Command completed
- a1 = `1`: Command spawned (not used by the XROM CLP)
- a1 = `2`: Command processing failed
- a1 = `3`: Command execution failed
- ah: Error code that indicates the reason for a failure

Table 3: CLP Completion Codes

	CLP Error Return Values (Refer to DMTF SM CLP Tables 6, 9, and 11)		
Error Type	**a1**	**ah**	**Description**
Success status	0	0	Command completed successfully
CLP input string longer than 256 characters (255+null)	2	252	Command syntax error
CLP verb is unsupported or unrecognized	2	255	Command not recognized error
Missing space between CLP words	2	252	Command syntax error
Options parameter present in CLP string after CLP verb	2	247	Command options not supported
Target name wrong (for example, `<context>`)	2	246	Invalid context error
Unknown or unsupported command (parameter string)	2	246	Invalid context error
Physical link number out of range or invalid (for example, in `fcportx`, *x* is invalid)	2	246	Invalid context error
Error writing to flash or Error in CLP code that updates the flash parameter	3	0	EAX (bits 31:16) = 2 Adapter card error(2)

Revision History

ELXCLP-SP118-100; July 26, 2021

Initial release.


BROADCOM®