Emulex® Elxflash and LpCfg Utilities

User Guide
Release 14.4

Broadcom reserves the right to make changes without further notice to any products or data herein to improve reliability, function, or design. Information furnished by Broadcom is believed to be accurate and reliable. However, Broadcom does not assume any liability arising out of the application or use of this information, nor the application or use of any product or circuit described herein, neither does it convey any license under its patent rights nor the rights of others.

Table of Contents

Chapter 1: Introduction 7

1.1 Elxflash Offline Kit 7

1.2 Elxflash Utility 7

1.3 Command Syntax 8

1.4 Supported Platforms 8

1.5 Abbreviations 8

Chapter 2: Installation 10

2.1 Platform Prerequisites 10

2.2 Installing the Elxflash Offline Kit for Linux 10

2.2.1 Updating an Elxflash Offline Kit for Linux 10

2.2.2 Uninstalling the Elxflash Offline Kit for Linux 11

2.3 Installing the Elxflash Offline Kit for Windows PE 11

2.3.1 Updating an Elxflash Offline Kit for Windows PE 11

2.3.2 Uninstalling the Elxflash Offline Kit for Windows PE 11

2.4 Extracting the Driver Files from the Elxflash Offline Kit for Windows 11

2.5 Extracting the Elxflash Stand-alone Kit for Linux, Windows, and VMware 12

Chapter 3: Elxflash Utility Command Line Interface 13

3.1 Using the fwmatrix.txt File Mode 13

3.2 Using the Autodiscovery Mode 13

3.3 Autodiscovery (/auto) 14

3.4 Downgrade (/downgrade or /g) 14

3.5 Force Firmware(/f) 15

3.6 Force Firmware (/ff) 15

3.7 Firmware Matrix Directory Change (/fmd) 15

3.8 Help (/h or /?) 15

3.9 Image Directory Change (/id) 16

3.10 Image Version Display (/iv) 16

3.11 Log (/log) 16

3.12 Preview (/p) 17

3.13 Query (/q) 17

3.14 Ramdrive (/ramdrive) 18

3.15 Rewrite (/rewrite or /e) 18

3.16 Silent (/s) 18

3.17 Update (/update) 19

3.18 Verbose (/v) 19

3.19 VPD (/vpd) 19

3.20 XML Output (/xml) 20

Chapter 4: LpCfg Utility Command Line Interface 21
4.1 Running the LpCfg Utility from the Command Prompt 21
4.2 Running the LpCfg Utility Included with the Elxflash Stand-alone Kit 22
4.2.1 Linux and VMware 22
4.2.2 Windows 23
4.3 Supported Commands 24
4.3.1 Help Command and General Adapter Management Commands 25
4.3.1.1 Viewing the Syntax for Commands (help) 25
4.3.1.2 Resetting an Adapter (reset) 25
4.3.1.3 Resetting an Adapter to the Factory Defaults (factorydefaults) 26
4.3.2 Viewing Adapter Information 26
4.3.2.1 Viewing the LpCfg Utility Version Information (version) 26
4.3.2.2 Viewing the VPD (vpd) 26
4.3.2.3 Viewing Boot Code Versions (listboot) 26
4.3.2.4 Viewing Adapter Attributes (hbaattr) 27
4.3.2.5 Viewing All Adapters in the System (listhba) 27
4.3.2.6 Viewing the WWN of All Adapters in the System (listwwn) 27
4.3.2.7 Viewing Firmware Program Revisions (listrev) 27
4.3.2.8 Viewing Selected Configuration Regions (readconfig) 28
4.3.2.9 Viewing the Flash Device (getflashpart) 28
4.3.3 Firmware Commands 28
4.3.3.1 Viewing Firmware Parameters (getfwparams) 28
4.3.3.2 Setting Firmware Parameters (setfwparam) 28
4.3.4 Firmware Download Commands 29
4.3.4.1 Downloading a File (download) 29
4.3.5 World Wide Name Commands 30
4.3.5.1 Saving WWN Data to a File (savewwn) 30
4.3.5.2 Restoring the WWN and Updating the NVPARMS (restorewwn) 30
4.3.5.3 Restoring the NVPARMS (restorenvwwn) 30
4.3.5.4 Restoring the IEEE Address (restoredefwwn) 31
4.3.6 Boot Code Commands 31
4.3.6.1 Configuring Boot Parameters (setbootparam) 32
4.3.6.2 Getting Boot Parameters (getbootparams) 37
4.3.7 Diagnostic Tests 38
4.3.7.1 Running the External Loopback Test (extloopback) 38
4.3.7.2 Running the Internal Loopback Test (intloopback) 39
4.3.7.3 Running the PCI Loopback Test (pciloopback) 39
4.3.7.4 Viewing Transceiver Data (getxcvrdata) 40

4.3.8 Using Script Files 40

4.3.8.1 Repeating a Series of Commands (repeat) 41

4.3.8.2 Enabling or Disabling Test Messages on the Screen (screendisplay)....41

4.3.8.3 Updating the Nonvolatile WWNN (scriptwwnn) 41

4.3.8.4 Updating the Nonvolatile WWPN (scriptwwpn)....42

4.3.8.5 Updating the Volatile WWNN (scriptvwwnn)....43

4.3.8.6 Updating the Volatile WWPN (scriptvwwpn)....43

4.3.8.7 Creating a Log (logfile) 44

Chapter 5: ESXi 8.0 Functionality 45

5.1 Elxflash Utility 45

5.1.1 Using the fwmatrix.txt File Mode....45

5.1.2 Using the Autodiscovery Mode 46

5.1.3 Autodiscovery (auto)....46

5.1.4 Downgrade (downgrade) 46

5.1.5 Force Firmware (forcefirmware)....47

5.1.6 Default Firmware Directory (-d <absolute path>)....47

5.1.7 Help....48

5.1.8 Image Version Display (imageversion -i <image_file>)....48

5.1.9 Preview (-p 1)....48

5.1.10 Query (-q 1)....49

5.1.11 Rewrite (rewrite)....49

5.1.12 Update (update)....50

5.1.13 VPD (-m vpd) 50

5.2 LpCfg Utility....50

5.2.1 Supported Commands....51

5.2.2 Help Command and General Adapter Management Commands 51

5.2.2.1 Viewing the Syntax for Commands (help) 51

5.2.2.2 Resetting an Adapter (reset)....52

5.2.2.3 Resetting an Adapter to the Factory Defaults (factorydefaults)....52

5.2.3 Viewing Adapter Information....53

5.2.3.1 Viewing the LpCfg Utility Version Information (version)....53

5.2.3.2 Viewing the VPD (vpd)....53

5.2.3.3 Viewing Adapter Attributes (hbaattr)....53

5.2.3.4 Viewing All Adapters in the System (listhba) 53

5.2.3.5 Viewing the WWN of All Adapters in the System (listwwn)....54

5.2.3.6 Viewing Firmware Program Revisions (listrev) 54

5.2.3.7 Viewing Selected Configuration Regions (readconfig) 54

5.2.3.8 Viewing the Flash Device (getflashpart) 54

5.2.4 Firmware Commands 55

5.2.4.1 Viewing Firmware Parameters (getfwparams) 55

5.2.4.2 Setting Firmware Parameters (setfwparam) 55

5.2.5 Firmware Download Commands 55

5.2.5.1 Downloading a File (download) 55

5.2.6 World Wide Name Commands 56

5.2.6.1 Saving WWN Data to a File (savewwn) 56

5.2.6.2 Restoring the WWN and Updating the NVPARMS (restorewwn) 56

5.2.6.3 Restoring the NVPARMS (restorenvwwn) 57

5.2.6.4 Restoring the IEEE Address (restoredefwwn) 57

5.2.6.5 Boot Code Commands 58

5.2.6.6 Configuring Boot Parameters (setbootparam) 58

5.2.6.7 Getting Boot Parameters (getbootparams) 63

5.2.7 Diagnostic Tests 64

5.2.7.1 Running the External Loopback Test (extloopback) 64

5.2.7.2 Running the Internal Loopback Test (intloopback) 65

5.2.7.3 Running the PCI Loopback Test (pciloopback) 65

5.2.7.4 Viewing Transceiver Data (getxcvrdata) 66

5.2.8 Using Script Files 66

5.2.8.1 Updating the Nonvolatile WWNN (scriptwwnn) 66

5.2.8.2 Updating the Nonvolatile WWPN (scriptwwpn) 67

5.2.8.3 Updating the Volatile WWNN (scriptvwwnn) 68

5.2.8.4 Updating the Volatile WWPN (scriptvwwpn) 69

Chapter 6: Status Messages and Error Codes 70

6.1 Elxflash Utility Status Messages 70

6.2 LpCfg Error Codes 71

Chapter 7: Troubleshooting 74

7.1 Unsupported Drivers 74

7.2 Erroneous Output 74

7.3 Firmware Download Failures 74

Chapter 1: Introduction

This document explains how to use the Emulex® Elxflash and LpCfg adapter management utility kits for Linux, Windows, and VMware® preboot environments.

1.1 Elxflash Offline Kit

The Elxflash Offline kit allows you to configure Emulex adapters before you install or boot a server operating system. You can also use the Elxflash Stand-alone kit, which lets you update firmware on Emulex adapters without installing any applications. See Section 4.2, Running the LpCfg Utility Included with the Elxflash Stand-alone Kit, for more information.

The Elxflash Offline kit includes the following items:

- linlpcfg and elxflash for Linux
- winlpcfg and elxflash for Windows

The Elxflash Stand-alone kit includes the following items:

- linlpcfg and elxflash for Linux
- linlpcf and elxflash for VMware
- winlpcfg and elxflash for Windows

The linlpcfg and the winlpcfg are management utilities for adapter configuration and diagnostics for Linux and Windows. The LpCfg utility can also download firmware, but you must manually select the adapters to be updated.

Elxflash is a firmware download utility. When the /auto switch is used, Elxflash automatically chooses the best image to download. When the `fwmatrix.txt` file is used, you choose the images to apply to the adapters.

NOTE: The offline utility must be installed on your system before you can run it. While the stand-alone utility does not need to be installed before it is run, you must go to the directory into which the stand-alone utility was extracted and run it from the command prompt.

See Chapter 4, LpCfg Utility Command Line Interface, for further information about the offline utility.

1.2 Elxflash Utility

The Elxflash utility uses a command line interface, which allows you to build scripts for automated and unattended firmware download solutions for the following Emulex FC adapters in production systems:

- LPe38000-series adapters
- LPe37000-series adapters
- LPe36000-series adapters
- LPe35000-series adapters
- LPe32000-series adapters
- LPe31000-series adapters

You can also download firmware on local and remote machines simultaneously.

The Elxflash Stand-alone kit, which allows you to update firmware on Emulex adapters without installing any applications, is also available. See Section 4.2, Running the LpCfg Utility Included with the Elxflash Stand-alone Kit, for more information.

The Elxflash utility uses autodiscovery (`/auto`), so you do not need to maintain the `fwmatrix.txt` configuration file. When autodiscovery is invoked, the utility automatically discovers local adapters and performs selected operations, such as `/ff` (force firmware), `/update`, `/downgrade`, and `/rewrite`.

When using autodiscovery, you must populate the firmware subdirectory with an image file for specific adapter models. Autodiscovery uses Emulex-defined firmware file names in the `firmware` subdirectory.

If you do not want to use autodiscovery, the Elxflash utility can update the firmware of an adapter using the `fwmatrix.txt` configuration file. When using the `fwmatrix.txt` file, you must update each supported adapter's type and firmware entry and place the corresponding firmware image into the `firmware` subdirectory. The `fwmatrix.txt` file enables all operations that are used with autodiscovery.

1.3 Command Syntax

In all cases, the commands are given with the prefix `os`. When you enter any command, you must replace `os` with either of the following prefixes:

- `./lin` – For the Linux LpCfg utility commands, for use on Linux machines.
- `win` – For the WinPE LpCfg utility commands, for use on WinPE machines.

The LpCfg utility commands are not case-sensitive, nor are their arguments. You can enter them in uppercase, lowercase, or mixed case. However, for Linux operating systems, use `linlpcfg` (all lowercase) for the utility commands; for example, `./linlpcfg listhba`.

The LpCfg utility commands use the following conventions:

- One space between the command name and the first argument.
- One space between additional arguments.

NOTE: Do not put a space before or after the equal sign within an argument.

WWN values are reported and specified with two 4-byte hexadecimal words: WWN word 0 (w0) and WWN word 1 (w1). To make the full WWN, concatenate the w0 and w1 values.

For detailed information on the command line interface, see Chapter 3, Elxflash Utility Command Line Interface, and Chapter 4, LpCfg Utility Command Line Interface.

1.4 Supported Platforms

For information on supported adapters and supported versions of operating systems and platforms, refer to the Elxflash utility release notes at www.broadcom.com.

1.5 Abbreviations

Table 1: Acronyms and Abbreviations

Acronym/Abbreviation	Description
BFS	boot from SAN
D_ID	destination identifier
FABL	fabric-assigned boot LUN
FCP	Fibre Channel Protocol
ms	milliseconds

Table 1: Acronyms and Abbreviations (Continued)

Acronym/Abbreviation	Description
NQN	NVMe qualified name
NVPARMS	nonvolatile parameters
PCI_ID	PCI device identification number
PE	Preinstallation Environment
PLOGI	Port Login
RPM	Red Hat Package Manager
SFS	SAN foundation software
VPD	vital product data
VWWNN	virtual WWNN
VWWPN	virtual WWPN
XML	Extensible Markup Language

Chapter 2: Installation

This chapter details the prerequisites and procedures for installing and uninstalling the Elxflash Offline and Stand-alone kits for Linux, VMware 7.0, and Windows operating systems. It also describes the update procedure for each operating system.

NOTE: The information in this section does not apply to ESXi 8.0. For ESXi 8.0 support, see Chapter 5.

2.1 Platform Prerequisites

The utilities included in the Elxflash Offline and Stand-alone kits for Linux have installation prerequisites that must be met before the utilities are installed. There are no prerequisites for the Elxflash Offline and Stand-alone kits on the VMware and Windows operating systems.

The Linux platform uses additional software, such as libraries, to run the different versions of the utility. This section identifies the prerequisites.

NOTE: The Linux driver must be installed on the system for the applications to work properly.

The following software must be installed to run the utility:

- libnl
- lspci
- libsysfs (optional)

2.2 Installing the Elxflash Offline Kit for Linux

The Elxflash Offline kit for Linux uses an install script to install the elxflash and linlpcfg utility RPM packages. The install script determines the correct architecture and distribution, and it updates the existing Elxflash Offline and Elxlinlpcfg RPM packages. If there are no existing Elxflash Offline or Elxlinlpcfg RPM packages, the install script installs the packaged Elxflash Offline and Elxlinlpcfg RPM packages.

To install the Elxflash Offline kit for Linux, perform the following steps:

1. Untar the installation tarball.
2. Run the install script located on the root of the installation kit.

Example:

```
$ tar zxvf elxflashOffline-linux-<version>-<rel>.tgz
$ cd elxflashOffline-linux-<version>-<rel>
$ ./install.sh
```

2.2.1 Updating an Elxflash Offline Kit for Linux

To update an existing Elxflash Offline kit for a Linux installation, run the `install.sh` script to update the Elxflash Offline and Elxlinlpcfg RPM packages.

2.2.2 Uninstalling the Elxflash Offline Kit for Linux

The Elxflash Offline kit for Linux uses an uninstall script to uninstall the utility. The uninstall script performs specific actions depending on the switches that are used.

- `./uninstall.sh` – Uninstalls the Elxflash Offline and Elxlinlpcfg RPM packages.
- `./uninstall.sh -h` – Displays a summary of all available switches.

To uninstall the Elxflash Offline kit for Linux, run the uninstall script located on the root of the installation kit.

Example:

```
$ cd elxflashOffline-linux-<version>-<rel>
$ ./uninstall.sh
```

2.3 Installing the Elxflash Offline Kit for Windows PE

To install the Elxflash Offline kit for Windows PE, perform the following steps:

1. Unzip the `Offline-WinPE-<version>-<rel>.zip` file.
2. Change the directory to the correct architecture subdirectory.
3. Run `setupElxAll-<arch>.exe`.

The following components are installed:

- Storport FC driver
- winLpCfg – Elxflash Offline

NOTE: The `setupElxAll-<arch>.exe` file is compatible only with WinPE. It does not function with any version of Windows Server.

2.3.1 Updating an Elxflash Offline Kit for Windows PE

To update an existing Elxflash Offline kit installation, run the same steps that you previously used for installing the Elxflash Offline kit for Windows PE.

The installer uninstalls the existing version and then installs the updated version.

2.3.2 Uninstalling the Elxflash Offline Kit for Windows PE

To uninstall the Elxflash Offline kit for Windows PE, run the following commands:

```
cd \ProgramData\Tarma Installer\
setup.exe /remove
```

2.4 Extracting the Driver Files from the Elxflash Offline Kit for Windows

To extract the driver files from the Elxflash Offline kit, you can use the installer or the command line. When extracting these files, all files, both x64 and x86, are extracted to `<Drive>:\My Documents\Emulex\Drivers`. This location is hard-coded and cannot be changed.

NOTE: Extracting all the drivers takes an additional 28 MB. By default, Windows PE allocates only 32 MB of writable memory. The size of the writable memory can be increased when building the WinPE ISO image.

You can extract the driver files using the GUI mode by running the installer, or you can use the command line in silent mode.

To extract the drivers from the Elxflash Offline kit for Windows, perform the following steps:

- In GUI mode, perform the following steps:
  a. Run the installer.
  b. From the Installation Options screen, select **Extract All Drivers**, and deselect **Install elxApp_Drivers (64-bit)**.
  c. Click **Install** to continue.
- In silent mode, from the command line, type the following command:
  ```
  start/wait SetupElxAll-x64.exe/q2 extract=2
  ```

NOTE: Values for the extract parameter are:

- `0` – Install the package; do not extract the drivers (default).
- `1` – Install the package; extract the drivers.
- `2` – Do not install the package; extract the drivers.

2.5 Extracting the Elxflash Stand-alone Kit for Linux, Windows, and VMware

For Linux, ESXi 7.0, and Windows, the Elxflash Stand-alone kit is not installed; you must extract it from its zip file. After you extract the utility, the following directories are created. The Linux executable files are extracted to the `lx\` directory, the ESXi executable files are extracted to the `esxi\` directory, and the Windows executable files are extracted to the `win\` directory.

- `firmware\`
- `lx\`
- `esxi\`
- `win\`

In Windows, for example, two directories are visible in the kit's root directory after you extract the stand-alone utility. The top-level firmware subdirectory is the default directory in which Elxflash looks for firmware images. Ensure that firmware images are located in this directory.

```
# dir

07/05/2012  07:02 PM    <DIR>          .
07/05/2012  07:02 PM    <DIR>          ..
07/02/2012  03:14 PM    <DIR>          firmware
07/02/2012  03:15 PM    <DIR>          win
```

The Elxflash Stand-alone kit for Linux has the same dependencies as the Elxflash Offline kit for Linux. The Elxflash Stand-alone kits for Windows and ESXi 7.0 do not have any dependencies. See Section 2.1, Platform Prerequisites.

NOTE: Update and uninstall procedures for the Elxflash Stand-alone kit are not needed. When a new version is available, you extract it, replacing the older version.

NOTE: To manage FC HBAs, you must install the operating-system-specific FC drivers.

Chapter 3: Elxflash Utility Command Line Interface

NOTE: The information in this section does not apply to ESXi 8.0. For ESXi 8.0 support, see Chapter 5.

Firmware images are available on the Broadcom® support site at www.broadcom.com. For all adapters, firmware image files are expected to be in the subdirectory named `firmware`.

Two supported modes for updating firmware are available for each Elxflash utility switch. The `fwmatrix.txt` file mode allows you to specify which adapter models and firmware file to update. Autodiscovery mode automatically updates the discovered adapters with the contents of the `firmware` subdirectory.

Regardless of the firmware update mode you choose, use the `/query` (`/q`) switch with all operational switches.

For example:

```
# ./elxflash /auto /q
```

The following information is returned:

```
HBA=LPe35002, Port Type=FC, WWN=10:00:00:90:FA:94:94:91, PCI ID=F400, VID=10DF, SSID=F402, SVID=10DF,
Firmware=14.0.230.6, FLASH PART=MICRON
HBA=LPe35002, Port Type=FC, WWN=10:00:00:90:FA:94:94:92, PCI ID=F400, VID=10DF, SSID=F402, SVID=10DF,
Firmware=14.0.230.6, FLASH PART=MICRON
```

If the `/q` switch is used with an operational switch, the results include an additional field called `Supported Firmware`, which indicates the firmware image Elxflash will attempt to download.

Regardless of the mode and operational switch you choose, the following download summary is displayed after the command is executed:

```
<date><time>
HBA=<model>, Port Type=<port_type>, WWN=<wwn>,
Image=<image>, New=<version>, Old=<version>, Status=<description>
Return Code=<n>
```

where `<description>` is `Success` or `Error`, and `<n>` = `0` for completion with no errors or a nonzero error code for any error.

3.1 Using the fwmatrix.txt File Mode

To use the `fwmatrix.txt` file, you must update the `firmware` subdirectory with the appropriate firmware images. The `fwmatrix.txt` file includes adapter entries followed by a firmware image file entry.

NOTE: To avoid a firmware update failure, ensure that there are no spaces in the model names in the `fwmatrix.txt` file.

When the Elxflash utility uses the `fwmatrix.txt` file, it locates the matching adapter model entry and then attempts to download the specified firmware image file. If there are multiple entries, the utility uses only the last matching entry for the adapter that is being evaluated.

3.2 Using the Autodiscovery Mode

If the `/auto` switch is used with an operational switch, such as `/update`, the Elxflash utility automatically discovers adapters. Using the `firmware` subdirectory, the utility performs the operation specified by the switch on each adapter.

3.3 Autodiscovery (/auto)

Usage: `/auto`

The `/auto` switch instructs the Elxflash utility to ignore the `fwmatrix.txt` file, to automatically discover local adapters, and to perform specified operations employing an additional switch using the `firmware` subdirectory.

The `/auto` switch must be used with an additional operational switch, such as one of the following switches:

- `/downgrade`
- `/f`
- `/ff`
- `/rewrite`
- `/update`

Example usage:

`./elxflash /auto /update /q` – Updates the firmware using the `firmware` subdirectory.

- The `fwmatrix.txt` file is ignored. The desired versions of firmware must be in the `firmware` subdirectory.
- Using the `firmware` subdirectory, the Elxflash utility automatically discovers the best matching firmware for each installed and supported adapter.
- If multiple versions of firmware are found for an adapter, the Elxflash utility uses the most recent version when performing the firmware update.

3.4 Downgrade (/downgrade or /g)

Usage: `/downgrade or /g`

The `/downgrade` switch downgrades the firmware of each adapter if the currently installed versions are more recent than the downgrade versions. This switch cannot be used with the `/update` and `/rewrite` commands.

Example usage:

`./elxflash /downgrade /auto /q` – Downgrades the firmware using autodiscovery.

- The `fwmatrix.txt` file is ignored. The desired downgrade versions of firmware must be in the `firmware` subdirectory.
- If the downgrade versions are older than the currently installed versions on the adapter, the downgrade versions are downloaded to the adapter.
- If multiple downgrade versions of firmware are found for an adapter, the next-previous downgrade versions are downloaded to the adapter.

`./elxflash /downgrade /q` – Downgrades the firmware using the `fwmatrix`.txt file.

- For each installed and supported adapter, the current firmware version is compared to the version specified in `fwmatrix.txt`.
- If the downgrade version in `fwmatrix.txt` is older than the currently installed version, the downgrade version of firmware is downloaded to that adapter.

3.5 Force Firmware(/f)

Usage: `/f`

The `/f` switch forces a firmware download to an adapter regardless of the current version on the adapter, and it is performed regardless of any additional operational switches given on the command line. Also, see Section 3.6, Force Firmware (/ff).

Example usage:

`./elxflash /f /auto /q` – Forces a firmware download using autodiscovery.

- The `fwmatrix.txt` file is ignored. The desired versions of firmware must be in the `firmware` subdirectory.
- If multiple versions of firmware are found for an adapter, the Elxflash utility uses the most recent version when performing the firmware download.

`./elxflash /f /q`– Forces a firmware download using the `fwmatrix.txt` file.

- For each installed and supported adapter, this command forces a download of firmware using the versions specified in the `fwmatrix.txt` file.

3.6 Force Firmware (/ff)

Usage: `/ff`

The `/ff` switch forces a firmware download to an adapter regardless of the firmware version installed on the adapter or any additional operational switches given on the command line.

Example usage:

`./elxflash /ff /auto /q` – Forces a firmware download using autodiscovery.

- The `fwmatrix.txt` file is ignored. The desired versions of firmware must be in the `firmware` subdirectory.
- If multiple versions of firmware are found for an adapter, the Elxflash utility uses the most recent version when performing the firmware download.

`./elxflash /ff /q` – Forces a firmware download using the `fwmatrix.txt` file.

- For each installed and supported adapter, this command forces a download of firmware using the firmware version specified in the `fwmatrix.txt` file.

3.7 Firmware Matrix Directory Change (/fmd)

Usage: `/fmd=<directory>`

The `/fmd` switch changes the location of the base directory in which `fwmatrix.txt` is located.

3.8 Help (/h or /?)

Usage: `/h` or `/?`

The `/h` switch displays a help message with instructions for using the Elxflash utility.

3.9 Image Directory Change (/id)

Usage: `/id=<image_directory>`

The `/id` switch specifies the location of the `firmware` subdirectory.

Example usage:

`./elxflash /f /auto /id=/tmp /q` – Forces a firmware download using autodiscovery.

- Elxflash looks for the `firmware` subdirectory in the `/tmp` directory.
- The `fwmatrix.txt` file is ignored. The desired versions of firmware must be in the `firmware` subdirectory.
- In this example, each firmware image file must be placed in the `/tmp/firmware` subdirectory.
- If multiple versions of firmware are found for an adapter, Elxflash uses the most recent versions when performing the firmware download.

`./elxflash /f /id/tmp /q` – Forces a firmware download using the `fwmatrix.txt` file.

- Elxflash looks for the `firmware` subdirectory in the `/tmp` directory.
- For each installed and supported adapter, a forced download of firmware occurs using the versions specified in the `fwmatrix.txt` file.
- In this example, the firmware file must be placed in the `/tmp/firmware` subdirectory.

NOTE: This switch was `/i` in previous versions of the Elxflash utility.

3.10 Image Version Display (/iv)

Usage: `/iv=<image_path>/<image_file>`

The `/iv` command shows the firmware image file’s version number. You must specify a path to the firmware image file for the command to decode the image file’s version.

Example usage:

```
./elxflash /iv=firmware/prism_A14.2.123.45.grp
```

The following information is returned:

```
14.2.123.45
```

3.11 Log (/log)

Usage: `/log=<log_file.txt>`

The `/log` switch appends the output of the Elxflash utility to a text file. The `/log` switch can be used with any switch.

3.12 Preview (/p)

Usage: `/p`

The `/p` switch provides a download preview of all adapters the Elxflash utility can update using either autodiscovery or the `fwmatrix.txt file`. The preview switch can be used with any of the operational switches, such as the following:

- `/downgrade`
- `/f`
- `/ff`
- `/rewrite`
- `/update`

When the preview switch is used, the Elxflash utility displays a download summary, but it does not actually perform the download.

NOTE: If the `/p` switch is used, the `Status=<`*`description`*`>` field displays `Preview`.

Each adapter's download preview displays the adapter's old and new image versions. The old image version represents the image version that is currently on the adapter. The new image version represents the image version the Elxflash utility would use during a download.

Example usage:

`./elxflash /p /auto /update` – Previews an upgrade of firmware using autodiscovery.

- The `fwmatrix.txt` file is ignored. The desired update versions of firmware must be in the `firmware` subdirectory.
- If the update versions are later than the currently installed versions on the adapter, the Elxflash utility provides a download preview for each adapter that can be updated.
- If multiple update versions of firmware are found for an adapter, the Elxflash utility provides a download preview using the most recent versions.

`./elxflash /p` – Previews an upgrade of firmware using the `fwmatrix.txt` file.

- For each installed and supported adapter, the current firmware versions are compared to the versions specified in the `fwmatrix.txt` file.
- If the update versions in the `fwmatrix.txt` file are more recent than the currently installed versions, the Elxflash utility provides a download preview of firmware for each adapter that can be updated.

3.13 Query (/q)

Usage: `/q`

The `/q` switch displays an adapter's model, WWN, PCI_ID, firmware version, and flash part type. Query can be used with any switch.

Example usage:

```
C:\elxflashStandalone-windows-14.2.123.45-1\win>elxflash.bat /q
```

The following information is returned:

```
HBA=<model>, Port Type=FC, WWN =<wwn>, Firmware=<version>, Flash Part=<type>
```

3.14 Ramdrive (/ramdrive)

NOTE: This command is available in the Windows offline utility only.

Usage: `/ramdrive=<`*`drive letter`*`>[:]`

The `/ramdrive` switch specifies the drive on which to create temporary files and log files. The default is drive X, which is the default drive in Windows PE.

3.15 Rewrite (/rewrite or /e)

Usage: `/rewrite` or `/e`

The `/rewrite` switch updates the firmware of each adapter if the installed versions are earlier than, or the same as, the rewrite versions. This switch cannot be used with `/update` or `/downgrade`.

Example usage:

`./elxflash /rewrite /auto` – Rewrites the firmware using autodiscovery.

- The `fwmatrix.txt` file is ignored. The desired rewrite versions of firmware must be in the `firmware` subdirectory.
- If the rewrite versions are more recent than, or the same as, the versions installed on the adapter, the rewrite versions are downloaded to the adapter.
- If multiple rewrite versions of firmware are found for an adapter, the most recent versions are downloaded to the adapter.

`./elxflash /rewrite` – Rewrites the firmware using the `fwmatrix.txt` file.

- For each installed and supported adapter, the current firmware version is compared to the version in the `fwmatrix.txt` file.
- If the currently installed versions are less than or equal to the rewrite versions in the `fwmatrix.txt` file, the rewrite versions of firmware are downloaded to the adapter.

3.16 Silent (/s)

Usage: `/s`

The `/s` switch prevents all output from being displayed.

Example usage:

`./elxflash /f /auto /s` – Forces a firmware download using autodiscovery and muting all output to `stdout`.

- The `fwmatrix.txt` file is ignored. The desired versions of firmware must be in the `firmware` subdirectory.
- If multiple versions of firmware are found for an adapter, Elxflash uses the most recent version when performing the firmware download.
- No output is printed to `stdout`.

`./elxflash /f /s` – Forces a firmware download using the `fwmatrix.txt` file and mutes all output to `stdout`.

- For each installed and supported adapter, a download of firmware is forced using the versions specified in the `fwmatrix.txt` file.
- No output is printed to `stdout`.

3.17 Update (/update)

Usage: `/update`

The `/update` switch updates the firmware of each adapter if the currently installed versions are older than the update versions. This switch cannot be used with `/downgrade` or `/rewrite`.

Example usage:

`./elxflash /update /auto /q` – Upgrades the firmware using autodiscovery.

- The `fwmatrix.txt` file is ignored. The desired update versions of firmware must be in the `firmware` subdirectory.
- If the update versions are newer than the currently installed versions on the adapter, the update versions are downloaded to the adapter.
- If multiple update versions of firmware are found for an adapter, the most recent version is downloaded to the adapter.

`./elxflash /update /q` – Upgrades the firmware using the `fwmatrix.txt` file.

- For each installed and supported adapter, the current firmware version is compared to the version specified in `fwmatrix.txt`.
- If the update versions in `fwmatrix.txt` are more recent than the currently installed version, the update versions of firmware are downloaded to that adapter.

3.18 Verbose (/v)

Usage: `/v`

The `/v` switch displays progress messages, and it can be used with any switch. When this switch is used, the following information is displayed:

- A download summary for all adapters that had successful or failed downloads
- A summary of unsupported adapters, if applicable
- A message for each adapter the Elxflash utility did not update

3.19 VPD (/vpd)

Usage: `/vpd`

The `/vpd` command displays the VPD model name for supported adapters. You must use the `/vpd` command with one of the following switches:

- `/downgrade`
- `/f`
- `/ff`
- `/query` or `/q`
- `/rewrite`
- `/update`
- `/xml`

NOTE: The `/vpd` command does not display any information if it is used without one of the preceding switches. If you attempt to use the `/vpd` command without one of the preceding switches, an error message is displayed.

If the `/vpd` command is used with the `fwmatrix.txt` file, the model name in the `fwmatrix.txt` file must match the VPD model name reported by Elxflash. Supported adapters include legacy FC adapters that support VPD.

3.20 XML Output (/xml)

Usage: `/xml`

The `/xml` switch displays utility output in XML format.

Chapter 4: LpCfg Utility Command Line Interface

NOTE: The information in this section does not apply to ESXi 8.0. For ESXi 8.0 support, see Chapter 5.

The LpCfg utility allows you to configure Emulex adapters before you install or boot a server operating system.

The LpCfg utility includes the following modules:

- linlpcfg
- winlpcfg

NOTE: Usage code and examples show `oslpcfg`. Note that the *os* designation changes to `./lin` if you are using LpCfg for Linux or VMware and to `win` if you are using LpCfg for Windows.

You can use the LpCfg utility to perform the following tasks:

- View information about an Emulex adapter
- Reset the adapter
- Return the adapter to its factory default settings
- Update firmware files
- Select a boot device
- Read and update WWNs
- Update configuration regions
- Run diagnostic tests
- Read and process script files

4.1 Running the LpCfg Utility from the Command Prompt

NOTE: The LpCfg utility is a command line utility and does not include a GUI.

To run the LpCfg utility from the command prompt, perform the following steps:

1. Boot the system with a supported operating system.
2. Start the LpCfg utility with a valid command or a valid script file name
   - To start the LpCfg utility from the command line, move to the directory where the executable file resides and type the following command:

     ```
     oslpcfg <valid command>
     ```

NOTE: Replace *os* with `./lin` or `win` as appropriate.

- For example, to start the Linux or VMware LpCfg utility with a `reset` command, type the following command:

  ```
  ./linlpcfg reset n=2
  ```

- For example, to start the Windows LpCfg utility with a `reset` command, type the following command:

  ```
  winlpcfg reset n=2
  ```

   - To start the LpCfg utility with a script file name, move to the directory where the LpCfg utility resides and type the following command:

     ```
     oslpcfg @<script file name>
     ```

     For example, to start the Windows LpCfg utility with `script1.txt` in the `c:\test` directory, type the following command:

     ```
     winlpcfg @C:\test\script1.txt
     ```

NOTE: To redirect screen output to a file, add <*filename*> at the end of each command. For example:

```
oslpcfg listboot n=1 >result.out
```

For more information on script files, see Section 4.3.8, Using Script Files.

4.2 Running the LpCfg Utility Included with the Elxflash Stand-alone Kit

The Elxflash Stand-alone kit allows you to run Elxflash and LpCfg without installing the utilities. No drivers are included in the Stand-alone kit. No applications are installed because the utility runs from inside the kit.

The Elxflash Stand-alone kit has the same capability as the Elxflash Offline kit, except you use the following scripts to run the utility:

- Linux and VMware – `linlpcfg.sh` and `elxflash.sh`
- Windows – `winlpcfg.bat` and `elxflash.bat`

4.2.1 Linux and VMware

NOTE: Running 32-bit applications on Linux x86_64 is not supported. The `.sh` scripts run only native executable files.

To run the utility in Linux and VMware, perform the following steps:

1. Extract the kit contents.
2. Change directory (`cd`) to `ElxflashStandalone-linux-<`*`version`*`>`.

   The following directories must be present:

   - `firmware\`
   - `lx\`
3. For Elxflash, copy the firmware images to the `firmware` subdirectory.

NOTE: For Elxflash, each operating system architecture directory includes an `fwmatrix.txt` file. You must use the `fwmatrix.txt` file in the directory that matches the current operating system architecture.

4. Change directory (`cd`) to the `lx` directory.

   The following directories and files must be present:

   - `x86_64\`
   - `elxflash.sh`
   - `linlpcfg.sh`

   The `elxflash.sh` script stops the Emulex HBA Manager services, runs the Elxflash utility, and restarts the Emulex HBA Manager services.

   The `linlpcfg.sh` script stops the Emulex HBA Manager services, runs the linlpcfg utility, and restarts the Emulex HBA Manager services before exiting.

NOTE: If the Emulex HBA Manager application is installed, the Emulex HBA Manager libraries are used by the Elxflash and linlpcfg utilities.

The `linlpcfg.sh` and `elxflash.sh` scripts call the native versions of Elxflash and linlpcfg. For example, on Linux x86_64, the 64-bit utilities are called.

The adapter being managed by the utilities included in the Elxflash Stand-alone kit must not be managed simultaneously by other Emulex utilities, including the Emulex HBA Manager application.

5. To update firmware on an adapter, type the following command:

   ```
   # ./elxflash.sh /auto up
   ```

   To display a list of HBAs, type the following command:

   ```
   # ./linlpcfg.sh  listhba
   ```

   Each time Elxflash or LpCfg is run, a log file is created. On Linux, these files are called `/var/log/clu/elxflash.log` and `/var/log/clu/linlpcfg.log`.

4.2.2 Windows

To run the utility in Windows, perform the following steps:

1. Extract the kit contents.
2. Change directory (`cd`) to `ElxflashStandalone-windows-<`*`version`*`>`.

   The following directories must be present:

   – `firmware\`

   – `win\`
3. For Elxflash, copy the firmware images to the `firmware` subdirectory.

NOTE: For Elxflash, each operating system architecture directory includes an `fwmatrix.txt` file. You must use the `fwmatrix.txt` file in the directory that matches the current operating system architecture.

4. Change directory (`cd`) to the `win` directory.

   The following directories and files must be present:

   – `win32\`

   – `x64\`

   – `elxflash.bat`

   – `winlpcfg.bat`

   The `elxflash.bat` script configures the environment, runs the Elxflash utility, and reverts any changes before exiting.

   The `winlpcfg.bat` script configures the environment, runs the winlpcfg utility, and reverts any changes before exiting.

NOTE: The `elxflash.bat` and `winlpcfg.bat` scripts call the native versions of `elxflash.exe` and `winlpcfg.exe`. For example, on Windows x64, the 64-bit utilities are called.

The adapter being managed by the utilities included in the Elxflash Stand-alone kit must not be managed simultaneously by other Emulex utilities, including the Emulex HBA Manager application.

If the Emulex HBA Manager application is installed, the Emulex HBA Manager libraries are used by the Elxflash and winlpcfg utilities.

5. To update firmware on an adapter in Windows, type the following command:

   ```
   # elxflash.bat /auto /up
   ```

   To display a list of HBAs in Windows, type the following command from the `win` directory:

   ```
   # winlpcfg.bat  listhba
   ```

   Each time Elxflash or LpCfg is run, a log file is created. On Windows, these files are called `C:\clu\log\elxflash.log` and `C:\clu\log\winlpcfg.log`.

4.3 Supported Commands

Table 2 lists all of the LpCfg commands that are supported on Emulex adapters on various platforms.

- ✓ indicates commands that are supported on both winlpcfg and linlpcfg.
- L indicates commands that are supported only on linlpcfg.

Table 2: Supported Commands for LpCfg

Commands	x64	IBM PowerPC® HBAs
Operating Systems	RHEL 8.8+ RHEL 9.2+	RHEL 8.8+ RHEL 9.2+
	SLES 12 SP4+ SLES 15 SP4+	SLES 12 SP4+ SLES 15 SP4+
	ESXi 7.0+	N/A
	WinPE 10.1 (Dell only)	N/A
`download`	✓	L
`extloopback`	✓	L
`factorydefaults`	✓	L
`getbootparams`	✓	L
`getflashpart`	✓	L
`getfwparams`	✓	L
`getxcvrdata`	✓	L
`h (Help)`	✓	L
`hbaattr`	✓	L
`intloopback`	✓	L
`listboot`	✓	L
`listhba`	✓	L
`listrev`	✓	L
`listwwn`	✓	L
`logfile`	✓	L
`pciloopback`	✓	L
`readconfig`	✓	L
`reset`	✓	L
`restoredefwwn`	✓	L
`restorenvwwn`	✓	L
`restorewwn`	✓	L
`savewwn`	✓	L
`screendisplay`	✓	L
`scriptvwwnn`	✓	L
`scriptvwwpn`	✓	L
`scriptwwnn`	✓	L
`scriptwwpn`	✓	L
`setbootparam`	✓	L
`setfwparam`	✓	L
`version`	✓	L
`vpd`	✓	L

4.3.1 Help Command and General Adapter Management Commands

Use the following commands to view the online help, reset the adapter, or reset the adapter to factory defaults.

4.3.1.1 Viewing the Syntax for Commands (help)

To view the syntax for all available commands, type the following command:

```
oslpcfg help
```

To view the syntax for a specific command, type the following command:

```
oslpcfg help <command>
```

For example, the following command:

```
oslpcfg help download
```

returns a response similar to the following:

- For Linux and VMware:
  ```
  download n=<adapter> i=<path/image_filename>
  ```
- For Windows:
  ```
  download n=<adapter> i=<path\image_filename>
  ```

4.3.1.2 Resetting an Adapter (reset)

This command resets a specific adapter or all adapters in the system.

ATTENTION: Do not execute a reset on an FC adapter that is being used to boot from SAN. The reset might cause a loss of connectivity to the SAN and possible loss of data. To reset an FC adapter, make sure the adapter is not currently being used to boot from SAN. There are two ways to perform this action:

- Move the adapter you want to reset to a non-boot-from-SAN host, and reset it from there.
- If the host with the reset target adapter is also hosting other boot-from-SAN adapters, carry out a boot from SAN using one of the other boot-from-SAN adapters. The intended target adapter can now be reset because it is not being actively used for boot from SAN.

To reset one adapter, type the following command:

```
oslpcfg reset n=<adapter_number>
```

To reset all adapters in the system, type the following command:

```
oslpcfg reset n=all
```

4.3.1.3 Resetting an Adapter to the Factory Defaults (factorydefaults)

This command returns the adapter to the factory default profile and configuration. A reboot is required for the changes to take effect.

NOTE: After a successful factory reset the adapter is offline. Perform an immediate reboot to complete the reset and return the adapter to full functionality.

ATTENTION: Do not restore the factory defaults on an FC adapter that is being used to boot from SAN. The `factorydefaults` command might cause a loss of connectivity to the SAN and possible loss of data. To restore the factory defaults on an FC adapter, make sure the adapter is not currently being used to boot from SAN. There are two ways to perform this action:

- Move the adapter on which you want to restore the defaults to a non-boot-from-SAN host, and perform the restore defaults command from there.
- If the host with the restore defaults target adapter is also hosting other boot-from-SAN adapters, carry out a boot from SAN using one of the other boot-from-SAN adapters. The factory defaults can now be restored on the intended target adapter because it is not being actively used for boot from SAN.

To return an adapter to its default settings, type the following command:

```
oslpcfg factorydefaults n=<adapter_number|all>
```

The following example returns adapter number 1 to its default settings.

```
oslpcfg factorydefaults n=1
```

4.3.2 Viewing Adapter Information

Use the following commands to view different kinds of adapter information, such as the VPD model name, boot device information, and adapter attributes.

4.3.2.1 Viewing the LpCfg Utility Version Information (version)

This command shows the LpCfg utility version information.

To view this information, type the following command:

```
oslpcfg version
```

4.3.2.2 Viewing the VPD (vpd)

This command shows the VPD of the adapter specified by its number.

To display the VPD, type of the following command:

```
oslpcfg vpd n=<adapter_number>
```

4.3.2.3 Viewing Boot Code Versions (listboot)

This command lists all the boot code versions that are loaded in the flash of the adapter.

To list boot code versions, type one the following command:

```
oslpcfg listboot n=<adapter_number>
```

The following example lists boot code versions that are loaded on adapter number 3:

```
oslpcfg listboot n=3
```

4.3.2.4 Viewing Adapter Attributes (hbaattr)

This command displays adapter information.

To list the adapter attributes for all installed adapters, type the following command:

```
oslpcfg hbaattr
```

To list the adapter attributes for one adapter, type the following command:

```
oslpcfg hbaattr n=<adapter_number>
```

NOTE: The `hbaattr` command output includes Firmware Status information. If a reboot is required to activate new firmware, Firmware Status displays the following text:

```
Reboot the system to activate new firmware.
```

If an optional reboot is required to enable a new feature, Firmware Status displays the following text:

```
The new firmware is activated. Some features require an optional reboot. Refer to the
Adapter's Firmware and Boot Code Release Notes for details.
```

For a list of features that require a reboot to be enabled, refer to the firmware and boot code release notes for the adapter.

4.3.2.5 Viewing All Adapters in the System (listhba)

This command lists all installed adapters in the system. Information includes the adapter number, the IEEE address assigned by the manufacturer, the functional firmware, the adapter type, and possible mailbox errors.

To list all adapters in the system, type the following command:

```
oslpcfg listhba
```

4.3.2.6 Viewing the WWN of All Adapters in the System (listwwn)

This command lists all adapters installed in the system and shows the factory-assigned WWN, the nonvolatile WWPN, and the WWNN used to identify an adapter in the SAN.

The factory-assigned WWN is an IEEE address that cannot be changed in the field. The nonvolatile WWN can be modified in the field and persists after a restart of the operating system. The full factory-assigned WWN and nonvolatile WWN are a concatenation of the two 8-character values (word 0 and word 1) that are shown for each. You can modify the nonvolatile WWPN and WWNN using the `scriptwwpn` and `scriptwwnn` commands.

If the system does not have any Emulex adapters installed, error code 45 is returned.

To show the WWN information, type the following command:

```
oslpcfg listwwn
```

4.3.2.7 Viewing Firmware Program Revisions (listrev)

This command shows the firmware versions in the adapter’s flash memory, specified by their numbers.

To show revisions, type the following command:

```
oslpcfg listrev n=<adapter_number>
```

The following example lists the firmware revision information for adapter number 3:

```
oslpcfg listrev n=3
```

4.3.2.8 Viewing Selected Configuration Regions (readconfig)

This command shows the contents of the selected configuration region up to the initialized length or the specified byte count (if the initialized length is less than the specified byte count). The following region numbers are valid:

- 0
- 8
- 9
- 10
- 14
- 16 (which is remapped to 32)
- 32

You must initialize the configuration region first by writing data to it.

To read a configuration, type the following command:

```
oslpcfg readconfig n=<adapter_number> r=<region_number> l=<byte_count>
```

The following example reads the configuration for adapter number 1, region 0, byte count 20:

```
oslpcfg readconfig n=1 r=0 l=20
```

4.3.2.9 Viewing the Flash Device (getflashpart)

This command displays the flash device that is installed on the adapter. To view the flash device information, type the following command:

```
oslpcfg getflashpart n=<adapter_number>
```

The following example displays the flash device information for adapter number 1:

```
oslpcfg getflashpart n=1
```

4.3.3 Firmware Commands

4.3.3.1 Viewing Firmware Parameters (getfwparams)

This command displays the topology parameter.

To view the topology parameter, type the following command:

```
oslpcfg getfwparams n=<adapter_number>
```

4.3.3.2 Setting Firmware Parameters (setfwparam)

This command configures the topology parameter. To set the topology, type the following command:

```
oslpcfg setfwparam n=<adapter_number> topology <value>
```

where `<value>` is one of the following values:

- `0`=Auto (loop first)
- `1`=Auto (point-to-point first)
- `2`=Loop
- `3`=Point-to-point

NOTE: Point-to-point and FC-AL (loop) topologies are supported on the LPe31000-series and LPe32000-series adapters, and on the LPe35000-series and LPe36000-series adapters at speeds up to and including 16Gb/s. Point-to-point is the only topology supported on the LPe35000-series and LPe36000-series adapters at speeds greater than 16Gb/s.

After the `setfwparam` command is issued, you must reset the adapter port to activate the new firmware parameter setting.

4.3.4 Firmware Download Commands

The following firmware download commands include a command to download a firmware and a command to access the flash device directly.

4.3.4.1 Downloading a File (download)

This command downloads a firmware file to a specific adapter.

The adapter name is the name that appears when you run the `listhba` command. For more information on the `listhba` command, see Section 4.3.2.5, Viewing All Adapters in the System (listhba).

To download a firmware image file to an adapter specified by its number, type the following command:

```
oslpcfg download n=<adapter_number> i=<firmware_image_filename>
```

The following example downloads the `prism_A14.0.123.45.grp` firmware file to adapter number 6; in this example, the firmware file is for an LPe35000 adapter:

```
oslpcfg download n=6 i=prism_A14.2.123.45.grp
```

To verify the reset or boot status of new firmware, issue the `hbaatrib` command and locate the `Firmware Status` information. See Section 4.3.2.4, Viewing Adapter Attributes (hbaattr), for more information.

In some cases, a firmware update requires either a firmware reset or a basic PCIe reset, depending on the features available in the new firmware. A firmware reset is performed automatically if it is needed, regardless of the operating system. A basic PCIe reset is also performed automatically, but only on the following Linux operating systems:

- SLES 12 SP4 and later
- SLES 15 SP4 and later
- RHEL 8.8 and later
- RHEL 9.2 and later

If a firmware reset or basic PCIe reset occurs when the firmware is downloaded, a message similar to the following appears:

```
Download and pci reset successfully completed
```

In some cases, a full reboot is required to activate new firmware or to enable a new feature. In that case, a message similar to one of the following messages appears after the firmware download is complete:

```
Download successfully completed. Please reboot the system to activate new firmware.
Download completed. Some features require an optional reboot. Refer to the Adapter's Firmware and Boot
Code Release Notes for details.
```

For a list of features that require a reboot to be enabled, refer to the firmware and boot code release notes for the adapter.

4.3.5 World Wide Name Commands

The following commands allow you to save WWN data to a file and restore WWN data while updating the NVPARMS. You can also restore the NVPARMS and the IEEE address.

4.3.5.1 Saving WWN Data to a File (savewwn)

This command reads the original words 0 and 1 of the IEEE address, installed by manufacturing, from configuration regions 16 (or 32) of the adapter (specified by its number), and it saves the configuration region information in the selected WWN file.

NOTE: The output of this command is a binary file. Use the `hexdump` utility to view its contents.

To save the WWN data to a file, type the following command:

```
oslpcfg savewwn n=<adapter_number> c=<wwn_filename>
```

The following example reads the configuration region information on adapter number 4 and saves it to the contents of the `ctwwn.sav` file:

```
oslpcfg savewwn n=4 c=ctwwn.sav
```

4.3.5.2 Restoring the WWN and Updating the NVPARMS (restorewwn)

This command restores words 0 and 1 of the IEEE address from a specified file created with the `savewwn` command and uses them to update the NVPARMS port name with this IEEE address.

ATTENTION: Do not run a `restorewwn` command on an FC adapter that is being used to boot from SAN. The `restorewwn` command might cause a loss of connectivity to the SAN and possible loss of data. To restore the WWN and update NVPARMS on an FC adapter, make sure the adapter is not currently being used to boot from SAN. There are two ways to perform this action:

- Move the adapter on which you want to use the `restorewwn` command to a non-boot-from-SAN host, and run `restorewwn` from there.
- If the host with the `restorewwn` target adapter is also hosting other boot-from-SAN adapters, carry out a boot from SAN using one of the other boot-from-SAN adapters. You can now restore the WWN and update the NVPARMS on the intended target adapter because it is not being actively used for boot from SAN.

To restore the WWN, type the following command:

```
oslpcfg restorewwn n=<adapter_number> c=<wwn_filename>
```

The following example updates the NVPARMS on adapter number 4 with the `ctwwn.sav` file.

```
oslpcfg restorewwn n=4 c=ctwwn.sav
```

4.3.5.3 Restoring the NVPARMS (restorenvwwn)

ATTENTION: Do not run a `restorenvwwn` command on an FC adapter that is being used to boot from SAN. The `restorenvwwn` command might cause a loss of connectivity to the SAN and possible loss of data. To restore the NVPARMS on an FC adapter, make sure the adapter is not currently being used to boot from SAN. There are two ways to perform this action:

- Move the adapter on which you want to use the `restorenvwwn` command to a non-boot-from-SAN host, and run the command from there.
- If the host with the `restorenvwwn` target adapter is also hosting other boot-from-SAN adapters, carry out a boot from SAN using one of the other boot-from-SAN adapters. You can now restore the NVPARMS on the intended target adapter because it is not being actively used for boot from SAN.

This command restores the nonvolatile WWPN and WWNN to the adapter, replacing any volatile WWPN and WWNN data, without powering off the adapter.

If the adapter does not have firmware that supports the volatile WWN, the following error message appears:

```
Write Volatile Parms Error. Reported Error 48
```

If this error occurs, install firmware that supports the volatile WWN.

To restore the nonvolatile WWN, type the following command:

```
oslpcfg restorenvwwn n=<adapter_number>
```

The following example restores the nonvolatile WWN on adapter number 2.

```
oslpcfg restorenvwwn n=2
```

4.3.5.4 Restoring the IEEE Address (restoredefwwn)

This command reads the IEEE address (assigned by the manufacturer) and writes it to the nonvolatile WWPN and WWNN.

ATTENTION: Do not run a `restoredefwwn` command on an FC adapter that is being used to boot from SAN. The `restoredefwwn` command might cause a loss of connectivity to the SAN and possible loss of data. To restore the IEEE address on an FC adapter, make sure the adapter is not currently being used to boot from SAN. There are two ways to perform this action:

- Move the adapter on which you want to use the `restoredefwwn` command to a non-boot-from-SAN host, and run the command from there.
- If the host with the `restoredefwwn` target adapter is also hosting other boot-from-SAN adapters, carry out a boot from SAN using one of the other boot-from-SAN adapters. You can now restore the IEEE address on the intended target adapter because it is not being actively used for boot from SAN.

To restore the IEEE address, type the following command:

```
oslpcfg restoredefwwn n=<adapter_number>
```

The following example restores the IEEE address on adapter number 2.

```
oslpcfg restoredefwwn n=2
```

4.3.6 Boot Code Commands

To set the boot device with the offline utility, run the following commands in this order:

1. Use the `listboot` command to verify that the boot code is present. See Section , *os*lpcfg listboot n=3, for more information.

   Example:

   ```
   oslpcfg listboot n=1
   ```

   Sample return for an HBA with x86 BootBIOS enabled:

   ```
   Boot Code (enabled)          = 14.2.123.4
   fc_universal_bios_version    = 14.2.234.5
   fc_x86_bios_version          = 14.2.345.6
   fc_efi_bios_version          = 14.2.567.8
   Command completed, NO Error
   ```

Sample return for an HBA with x86 BootBIOS disabled:

```
Boot Code (disabled)         = 14.2.123.4
fc_universal_bios_version    = 14.2.234.5
fc_x86_bios_version          = 14.2.345.6
fc_efi_bios_version          = 14.2.567.8
Command completed, NO Error
```

2. Use the `setbootparam` command to perform the following tasks:
   – Enable boot code (`enableadapterboot` parameter).
   – Configure the HBA to boot from SAN (`enablebootfromsan` parameter).
   – Set alternate boot devices (`d_id`, `lun`, `targetwwpn`, or `targetid` parameter).
3. Run the `getbootparams` command (Section 4.3.6.2, Getting Boot Parameters (getbootparams)) to verify the configuration.
4. Configure the system BIOS so the adapter boot device is the highest in the boot order.

4.3.6.1 Configuring Boot Parameters (setbootparam)

This command changes the adapter parameters and the boot device parameters. You can change adapter parameters and boot device parameters for x86 boot and EFI boot.

NOTE: If you are changing an adapter parameter, omit the `bootdev` keyword and value; otherwise, an error is reported.

If you are changing a boot device parameter for x86 boot and EFI boot, you must provide the `bootdev` keyword and value.

Changes made to x86 boot settings are also applied to EFI boot settings, and changes made to EFI boot settings are also applied to x86 boot settings.

To configure an adapter parameter, type the following command:

```
oslpcfg setbootparam n=<adapter_number> <type> <param> <param_value>
```

To configure a boot device parameter, type the following command:

```
oslpcfg setbootparam n=<adapter_number> <type> <param> <param_value> bootdev <bootdev_value>
```

Parameters

`adapter_number`	The number of the adapter to be configured.
`type`	The supported boot type (provided in Table 3, Boot Parameter): `x86`: x86Boot `efi`: UEFIBoot
`param`	See Table 3, Boot Parameter.
`param_value`	See Table 3, Boot Parameter.
`bootdev_value`	Boot device entry number (`0` through `7`).

NOTE: OpenBoot is no longer supported. The feature will be disabled in a future release.

The following table provides the available boot parameters, their param_value values, and their supported boot types.

Table 3: Boot Parameter

Parameter Type	Parameter Name	Description	Supported Boot Types	param_value
Adapter	autoscan	The autoscan parameter allows you to set auto scan and enable the first device in the boot entry list to issue a Name Server Inquiry. The autoscan parameter is available only if none of the eight boot entries are configured to boot from DID or WWPN. The factory default is 0. If more than one adapter with the same PCI bus number exists in the system, and each has a boot drive attached, the first PCI-scanned adapter is the boot adapter.	x86	0=Disabled 1=First device 2=First LUN 0 device 3=First non-LUN 0 device
Adapter	autobootsectorenable	The autobootsectorenable parameter automatically defines the boot sector of the target disk for the migration boot process. This applies only to HP MSA1000 arrays. If no partition is on the target, the default boot sector format is 63 sectors. The default setting is 0.	x86	0=Disable 1=Enable

Table 3: Boot Parameter (Continued)

Parameter Type	Parameter Name	Description	Supported Boot Types	param_value
Adapter	boottargetscan	The boottargetscan parameter allows you to specify the method of scanning for boot targets. The following methods can be selected: ■ Boot path from NVRAM targets: A boot scan discovers only LUNs that are saved to the adapter's NVRAM. Select up to eight attached devices to use as potential boot devices. Limiting discovery to a set of eight selected targets can greatly reduce the time it takes for the EFIBoot driver to complete discovery. ■ Boot path from discovered targets: A boot scan discovers all devices that are attached to the FC port. Discovery can take a long time on large SANs if this option is selected. ■ Do not create a boot path. ■ EFI FC scan level: Setting an EFI FC scan level allows third-party software to toggle between a boot path from NVRAM and a boot path from discovered targets by manipulating a UEFI system NVRAM variable. ■ After the scan is set to an EFI FC scan level, the scan method can be changed without entering the UEFI boot configuration utility. You can establish the initial setting as either NVRAM targets (3) or discovered targets (4). ■ If you select one of the EFI FC scan levels settings, the scan is determined by the value of the EFIFCScanLevel variable maintained by the UEFI system firmware and boot code. The value of this variable can be changed either by using the bootargetscan parameter or by using third-party software.	efi	0=Boot path from NVRAM targets 1=Boot path from discovered targets 2=Do not create boot path 3=EFI FC scan level: NVRAM targets 4=EFI FC scan level: Discovered targets
Adapter	edd30enable	EDD 3.0 provides additional data to the operating system bootloader during the INT-13h function 48h (get device parameters) call. This information includes the path to the boot device and disk size. The default setting for EDD 3.0 is 1. **NOTE:** If EDD 3.0 is disabled, EDD 2.1 is enabled.	x86	0=Disable 1=Enable

Table 3: Boot Parameter (Continued)

Parameter Type	Parameter Name	Description	Supported Boot Types	param_value
Adapter	enableadapterboot	The enableadapterboot parameter enables or disables the boot code for the specified adapter. If the boot code is disabled, the adapter does not boot from the SAN, regardless of the value of the enablebootfromsan parameter. If the boot code is enabled, the adapter boots from the SAN if the enablebootfromsan parameter is also enabled.	x86 and efi	0=Disable 1=Enable
Adapter	enablebootfromsan	The enablebootfromsan parameter enables the adapter to boot from SAN. **NOTE:** The enableadadapterboot parameter must also be set to 1 for an adapter to boot from SAN. **NOTE:** To boot from SAN using NVMe, you must enable both the enablebootfromsan parameter and the enablenvmebootfromsan parameter.	x86 and efi	1=Disable 1=Enable
Adapter	enablefabl	FABL is no longer supported. The feature will be disabled in a future release.		
Adapter	envvarenable	The envvarenable parameter sets the boot controller order if the system supports the environment variable. The default setting is 0.	x86	0=Disable 1=Enable
Adapter	maxlunspertarget	The maxlunspertarget sets the maximum number of LUNs that are polled during device discovery.	efi	1 through 4096
Adapter	plogiretrytimer	The plogiretrytimer parameter is useful for Tachyon-based RAID arrays. In rare situations, a Tachyon-based RAID array resets itself and the port goes offline temporarily. When the port returns to operation, the PLOGI scans the loop to discover this device. The PLOGI retry interval is the time it takes for one PLOGI to scan the whole loop (if 126 AL_PAs are on the loop). Changes made to x86 are also applied to efi. Changes made to efi are also applied to x86.	x86 and efi	0=None 1=50 ms 2=100 ms 3=200 ms

Table 3: Boot Parameter (Continued)

Parameter Type	Parameter Name	Description	Supported Boot Types	param_value
Adapter	spinupdelayenable	The spinupdelayenable parameter allows you to enable or disable the disk spin-up delay. The factory default setting is 0. If at least one boot device has been defined, and the spin-up delay is enabled, the BIOS searches for the first available boot device. ■ If a boot device is present, the BIOS boots from it immediately. ■ If a boot device is not ready, the BIOS waits for the spin-up delay and, for up to 3 additional minutes, continues the boot scanning algorithm to find another multi-boot device. If boot devices have not been defined and the autoscan parameter is enabled, the BIOS waits for 5 minutes before scanning for devices. ■ In a private loop, the BIOS attempts to boot from the lowest target AL_PA it finds. ■ In an attached fabric, the BIOS attempts to boot from the first target found in the NameServer data.	x86	0=Disable 1=Enable
Adapter	startunitcommandenable	The startunitcommandenable parameter allows a start-unit command to be sent to a particular disk. This option might be needed if the disk you want to boot from is not yet spun up or started. You must know the specific LUN to which to issue the SCSI start unit command. The default setting is 0.	x86	0=Disable 1=Enable
Boot device	d_id	Specify the boot device by its device ID.	x86	Use the D_ID of the boot device (in decimal)
Boot device	lun	Specify the boot device by its starting LUN. Changes made to x86 are also applied to efi. Changes made to efi are also applied to x86.	x86 and efi	0 through 255
Boot device	targetwwpn	Specify the boot device by its target WWPN. Changes made to x86 are also applied to efi. Changes made to efi are also applied to x86.	x86 and efi	Use the target WWPN in the following format: nn:nn:nn:nn:nn:nn:nn:nn

Table 3: Boot Parameter (Continued)

Parameter Type	Parameter Name	Description	Supported Boot Types	param_value
NVMe adapter	`enablenvmebootfromsan`	Enable BFS on NVMe over FC. **NOTE:** To boot from SAN using NVMe, you must enable both the `enablebootfromsan` parameter and the `enablenvmebootfromsan` parameter.	`efi`	`0`=Disable `1`=Enable
NVMe boot device	`namespaceid`	Specify the boot device by its NVMe namespace ID value.	`efi`	`0` through `0xFFFFFFFF`
NVMe boot device	`nvmewwpn`	Specify the boot device by its NVMe WWPN.	`efi`	Use the NVMe WWPN in the following format: `nn:nn:nn:nn:nn:nn:nn:nn`
NVMe boot device	`nvmewwnn`	Specify the boot device by its NVMe WWNN.	`efi`	Use the NVMe WWNN in the following format: `nn:nn:nn:nn:nn:nn:nn:nn`
NVMe boot device	`subsystemnqn`	Specify the boot device by its subsystem NQN.	`efi`	Use the subsystem NQN string, which must conform to NQN standards. The maximum length is 233 characters.

4.3.6.2 Getting Boot Parameters (getbootparams)

This command displays the configured adapter parameters and boot device parameters.

To display boot parameters, type the following command:

```
oslpcfg getbootparams n=<adapter_number> <boot_type>
```

where *<boot_type>* is one of the following values:

- `x86`: x86 Boot BIOS
- `efi`: UEFI Boot

NOTE: OpenBoot is no longer supported. The feature will be disabled in a future release.

NOTE: All Emulex adapters do not support the same boot parameters. Boot parameters that are not supported by the adapter are not displayed.

Following is an example of the boot parameters for the x86 boot type on an LPe35000-series HBA:

```
WinLpCfg.exe getbootparams n=1 x86
Command: getbootparams n=1 x86

GetBootParams: Success

Parameter Name         Value
--------------         -----
 Standard Parameters
  EnableAdapterBoot        1   Enabled
  EnableBootFromSAN        1   Enabled
 Advanced Parameters
```

```
  AutoBootSectorEnable     0   Disabled
  AutoScan                 0   Disabled
  EDD30Enable              1   Enabled
  EnvVarEnable             0   Disabled
  PlogiRetryTimer          0   None
  SpinupDelayEnable        0   Disabled
  StartUnitCommandEnable   0   Disabled
 Boot Device Parameters
  Entry  Lun    D_ID        TargetWwpn
  -----  ---  ------  -----------------------
      0  000  000000  00:00:00:00:00:00:00:00
      1  000  000000  00:00:00:00:00:00:00:00
      2  000  000000  00:00:00:00:00:00:00:00
      3  000  000000  00:00:00:00:00:00:00:00
      4  000  000000  00:00:00:00:00:00:00:00
      5  000  000000  00:00:00:00:00:00:00:00
      6  000  000000  00:00:00:00:00:00:00:00
      7  000  000000  00:00:00:00:00:00:00:00
```

4.3.7 Diagnostic Tests

The following commands let you run diagnostic tests on your adapter.

4.3.7.1 Running the External Loopback Test (extloopback)

ATTENTION: Do not perform an `extloopback` test on an FC adapter that is being used to boot from SAN. The `extloopback` command might cause a loss of connectivity to the SAN and possible loss of data. To perform an `extloopback` command on an FC adapter, make sure the adapter is not currently being used to boot from SAN. There are two ways to perform this action:

- Move the adapter on which you want to perform the external loopback test onto a non-boot-from-SAN host, and perform the test from there.
- If the host with the test target adapter is also hosting other boot-from-SAN adapters, carry out a boot from SAN using one of the other boot-from-SAN adapters. The extloopback command can now be run on the intended target adapter because it is not being actively used for boot from SAN.

This command runs the external loopback test. You must put a loopback plug in each adapter port to be tested. You can test a specific adapter in the system. Specify the number of times you want the test to repeat, and direct the test response if an error is found.

NOTE: The `extloopback` command does not support testing all installed adapters using `n=all`.

To run the external loopback test, type the following command:

```
oslpcfg extloopback n=<adapter_number> r=<repeat_count> o=<option_on_error>
```

where:

- `r` = 1 to 4096
- `o` = 1 to 3
  - `1` stops the test on the first error.
  - `2` ignores three errors and either stops the test on the fourth error or continues testing the next adapter on the fourth error.
  - `3` ignores errors and continues the test.

The following example runs the external loopback test 50 times on adapter number 1 and stops the test if an error occurs:

```
oslpcfg extloopback n=1 r=50 o=1
```

4.3.7.2 Running the Internal Loopback Test (intloopback)

ATTENTION: Do not perform an `internal` loopback test on an FC adapter that is being used to boot from SAN. The `intloopback` command might cause a loss of connectivity to the SAN and possible loss of data. To perform an `intloopback` command on an FC adapter, make sure the adapter is not currently being used to boot from SAN. There are two ways to perform this action:

- Move the adapter on which you want to perform the internal loopback test onto a non-boot-from-SAN host, and perform the test from there.
- If the host with the test target adapter is also hosting other boot-from-SAN adapters, carry out a boot from SAN using one of the other boot-from-SAN adapters. The `intloopback` command can now be run on the intended target adapter because it is not being actively used for boot from SAN.

This command runs the internal loopback test. You can run the test on a specific adapter in the system. Specify the number of times you want the test to repeat, and indicate the action to be taken if an error occurs.

NOTE: The `intloopback` command does not support testing all installed adapters using `n=all`.

Do not use the internal loopback test on Emulex blade adapters.

To run the internal loopback test on an FC adapter, type the following command:

```
oslpcfg intloopback n=<adapter_number> r=<repeat_count> o=<option_on_error>
```

where:

- `r` = 1 to 4096
- `o` = 1 to 3
  - `1` stops the test on the first error.
  - `2` ignores three errors and either stops the test on the fourth error or continues testing the next adapter on the fourth error.
  - `3` ignores errors and continues the test.

The following example runs the internal loopback test 100 times on FC adapter number 1 and stops the test if an error occurs:

```
oslpcfg intloopback n=1 r=100 o=1
```

4.3.7.3 Running the PCI Loopback Test (pciloopback)

ATTENTION: Do not perform a PCI loopback test on an FC adapter that is being used to boot from SAN. The `pciloopback` command might cause a loss of connectivity to the SAN and possible loss of data. To perform a `pciloopback` command on an FC adapter, make sure the adapter is not currently being used to boot from SAN. There are two ways to perform this action:

- Move the adapter on which you want to perform the PCI loopback test onto a non-boot-from-SAN host, and perform the test from there.
- If the host with the test target adapter is also hosting other boot-from-SAN adapters, carry out a boot from SAN using one of the other boot-from-SAN adapters. The `pciloopback` command can now be run on the intended target adapter because it is not being actively used for boot from SAN.

This command runs the PCI loopback test. You can run the test on a specific adapter or on all adapters in the system. Specify the number of times you want the test to repeat, and indicate the action to be taken if an error occurs.

To run the PCI loopback test, type the following command:

```
oslpcfg pciloopback n=<all|adapter_number> r=<repeat_count> o=<option_on_error>
```

where:

- `r` = 1 to 4096
- `o` = 1 to 3
  - `1` stops the test on the first error.
  - `2` ignores three errors and either stops the test on the fourth error or continues testing the next adapter on the fourth error.
  - `3` ignores errors and continues the test.

The following example runs the PCI loopback test 100 times on FC adapter number 2 in the system and stops the testing if any errors occur.

```
oslpcfg pciloopback n=2 r=100 o=1
```

4.3.7.4 Viewing Transceiver Data (getxcvrdata)

This command displays transceiver data for an adapter port.

To view transceiver data, type the following command:

```
oslpcfg getxcvrdata n=<adapter_number>
```

4.3.8 Using Script Files

Use script files to efficiently perform tasks. Script files are common sequences of commands you use when performing tasks.

You can group commands together and run them using a script file. You can also enter comment lines, which begin with a semicolon. Each line follows the command syntax documented in this document. Using the offline utility, you can perform the following actions:

- Run commands entered in a script file. Use the `@` command to run the script file.
- Run commands multiple times. Add the `repeat` command as the last line of the script file.
- Create a log of test results. Add the `logfile` command as the first line of the script file.

To run a script file, type the following command:

```
oslpcfg @<scriptname.txt>
```

The following example runs the script script1.txt, which resides in the current directory and runs all the commands in that script file.

```
oslpcfg @script1.txt
```

NOTE: To interrupt and stop any script, press **S** on the keyboard.

The following is a sample script file. Each command follows the syntax covered previously. The comment lines begin with a semicolon (;).

```
version
screendisplay o=0
;download n=6 i=prism_A14.0.123.45.grp
;reset n=1 s=0
;reset n=2 s=0
reset n=all s=0
listboot n=1
setbootparam n=1 efi enableadapterboot 1
; pciloopback n=1 r=10 o=1
; pciloopback n=2 r=10 o=2
; pciloopback n=all r=50 o=3
intloopback n=1 r=10 o=1
intloopback n=2 r=10 o=1
extloopback n=1 r=40 o=3
extloopback n=2 r=40 o=3
;repeat r=10
```

4.3.8.1 Repeating a Series of Commands (repeat)

Enter the `repeat` command at the end of a script file to repeat a series of commands from the beginning of the script file a specific number of times.

NOTE: To interrupt and stop the `repeat` command, press **S** on the keyboard.

To repeat the series of commands in the script file, add the following as the last line of the file:

```
repeat r=<repeat_count>
```

The following example repeats the series of commands in the script file 10 times.

```
repeat r=10
```

4.3.8.2 Enabling or Disabling Test Messages on the Screen (screendisplay)

The `screendisplay` command enables or disables test message displays on the screen.

- `o=0` – Prevents messages from being displayed
- `o=1` – Enables messages

NOTE: This command is supported only in script files.

To enable or disable test message displays on the screen, add the following line to the script:

```
screendisplay o=<display_option>
```

In the following script file example, messages from the `version` and `listhba` commands appear in the log file and on the screen. After the `screendisplay` command is set to `0`, the resulting messages for all successive commands (`download`, `listboot`, and `enableadapterboot`) appear only in the log file, not on the screen.

```
version
listhba
screendisplay o=0
download n=6 i=c:\temp\prism_A14.0.123.45.grp
listboot n=2
setbootparam n=1 efi enableadapterboot 1
```

4.3.8.3 Updating the Nonvolatile WWNN (scriptwwnn)

ATTENTION: Do not run a `scriptwwnn` command on an FC adapter that is being used to boot from SAN. The `scriptwwnn` command might cause a loss of connectivity to the SAN and possible loss of data. To update the nonvolatile WWNN on an FC adapter, make sure the adapter is not currently being used to boot from SAN. There are two ways to perform this action:

- Move the adapter on which you want to use the `scriptwwnn` command to a non-boot-from-SAN host, and run the script from there.
- If the host with the target adapter is also hosting other boot-from-SAN adapters, carry out a boot from SAN using one of the other boot-from-SAN adapters. You can now run the script to update the nonvolatile WWNN on the intended target adapter because it is not being actively used for boot from SAN.

This command reads the WWNN words 0 and 1 from the command line to update the nonvolatile WWNN. You can also include this command in a script file. When the adapter is discovered, the new WWNN value is used. The adapter retains the original WWNN in another region of the firmware.

ATTENTION: Use the `scriptwwnn` command with caution. If you use the same WWNN on more than one adapter in a fabric, unpredictable results might occur.

To change WWNN words 0 and 1 from the command line, type the following command:

```
oslpcfg scriptwwnn n=<adapter_number> w0=<wwnn_word_0> w1=<wwnn_word_1>
```

The following example updates nonvolatile WWNN word 0 and word 1 for adapter number 1:

```
oslpcfg scriptwwnn n=1 w0=10000345 w1=B620A1B2
```

NOTE: Word 0 of WWNN and WWPN names must follow one of the following formats:

```
1 0 0 0 0 x x x
2 x x x x x x x
3 x x x x x x x
5 x x x x x x x
```

If the `scriptvwwnn` command has been used previously, the adapter continues to use that WWNN until you change the WWNN with the `restorenvwwn` command.

4.3.8.4 Updating the Nonvolatile WWPN (scriptwwpn)

ATTENTION: Do not run a `scriptwwpn` command on an FC adapter that is being used to boot from SAN. The `scriptwwpn` command might cause a loss of connectivity to the SAN and possible loss of data. To update the nonvolatile WWPN on an FC adapter, make sure the adapter is not currently being used to boot from SAN. There are two ways to perform this action:

- Move the adapter on which you want to use the `scriptwwpn` command to a non-boot-from-SAN host, and run the script from there.
- If the host with the target adapter is also hosting other boot-from-SAN adapters, carry out a boot from SAN using one of the other boot-from-SAN adapters. You can now run the script to update the nonvolatile WWPN on the intended target adapter because it is not being actively used for boot from SAN.

This command reads WWPN words 0 and 1 from the command line to update the nonvolatile WWPN. You can also include this command in a script file. When it is discovered, the adapter uses the new WWNN value. It does not use the original IEEE address assigned by manufacturing (located in configuration region 16 or 32).

ATTENTION: Use the `scriptwwpn` command with caution. If you use the same WWPN on more than one adapter in a fabric, unpredictable results might occur.

NOTE: If the `scriptvwwnn` command has been used previously, the adapter continues to use that WWPN until you change the WWPN with the `restorenvwwn` command.

To change WWPN words 0 and 1 from the command line, type the following command:

```
oslpcfg scriptwwpn n=<adapter_number> w0=<wwpn_word_0> w1=<wwpn_word_1>
```

The following example updates the nonvolatile WWPN word 0 word 1 for adapter number 1:

```
oslpcfg scriptwwpn n=1 w0=20A2D6B8 w1=C920A1B2
```

4.3.8.5 Updating the Volatile WWNN (scriptvwwnn)

ATTENTION: Do not run a `scriptvwwnn` command on an FC adapter that is being used to boot from SAN. The `scriptvwwnn` command might cause a loss of connectivity to the SAN and possible loss of data. To update the volatile WWNN on an FC adapter, make sure the adapter is not currently being used to boot from SAN. There are two ways to perform this action:

- Move the adapter on which you want to use the `scriptvwwnn` command to a non-boot-from-SAN host, and run the script from there.
- If the host with the target adapter is also hosting other boot-from-SAN adapters, carry out a boot from SAN using one of the other boot-from-SAN adapters. You can now run the script to update the volatile WWNN on the intended target adapter because it is not being actively used for boot from SAN.

This command reads the WWNN words 0 and 1 from the command line to update the volatile WWNN. The next time the adapter is discovered, it uses this new WWNN. It does not use the original IEEE address assigned by manufacturing (located in configuration region 16 or 32), nor does it use the value entered by the `scriptwwnn` commands.

If the adapter does not have firmware that supports the volatile WWN, the following error message appears:

```
Write Volatile Parms Error. Reported Error 48
```

If this error message appears, install firmware that supports the volatile WWN.

NOTE: Word 0 of WWNN and WWPN names must follow one of the following formats:

```
1 0 0 0 0 x x x
2 x x x x x x x
3 x x x x x x x
5 x x x x x x x
```

After you issue this command, the volatile WWNN is used by the adapter until the `restorenvwwn` command is issued or the system is restarted.

ATTENTION: Use the scriptvwwnn command with caution. If you use the same volatile WWNN on more than one adapter in a fabric, unpredictable results might occur.

To change volatile WWNN words 0 and 1 from the command line, type the following command:

```
oslpcfg scriptvwwnn n=<adapter_number> w0=<wwnn_word_0> w1=<wwnn_word_1>
```

The following example updates the volatile WWNN word 0 and word 1 for adapter number 1:

```
oslpcfg scriptvwwnn n=1 w0=20A2D6B8 w1=C920A1B2
```

4.3.8.6 Updating the Volatile WWPN (scriptvwwpn)

ATTENTION: Do not run a `scriptvwwpn` command on an FC adapter that is being used to boot from SAN. The `scriptvwwpn` command might cause a loss of connectivity to the SAN and possible loss of data. To update the volatile WWPN on an FC adapter, make sure the adapter is not currently being used to boot from SAN. There are two ways to perform this action:

- Move the adapter on which you want to use the `scriptvwwpn` command to a non-boot-from-SAN host, and run the script from there.
- If the host with the target adapter is also hosting other boot-from-SAN adapters, carry out a boot from SAN using one of the other boot-from-SAN adapters. You can now run the script to update the volatile WWPN on the intended target adapter because it is not being actively used for boot from SAN.

This command reads the WWPN words 0 and 1 from the command line to update the volatile WWPN. The next time the adapter is discovered, it uses this new WWPN. It does not use the original IEEE address assigned by manufacturing (located in configuration region 16 or 32), nor does it use the value entered by the `scriptwwpn` commands.

If the adapter does not have firmware that supports the volatile WWPN, the following error message appears:

```
Write Volatile Parms Error. Reported Error 48
```

If this error message appears, install firmware that supports the volatile WWPN.

NOTE: Word 0 of WWNN and WWPN names must follow one of the following formats:

```
1 0 0 0 0 x x x
2 x x x x x x x
3 x x x x x x x
5 x x x x x x x
```

After you issue this command, the volatile WWNN is used by the adapter until the `restorenvwwn` command is issued or the system is restarted.

ATTENTION: Use the `scriptvwwpn` command with caution. If you use the same VWWNN on more than one adapter in a fabric, unpredictable results might occur.

To change VWWPN words 0 and 1 from the command line, type the following command:

```
oslpcfg scriptvwwpn n=<adapter_number> w0=<wwpn_word_0> w1=<wwpn_word_1>
```

The following example updates the volatile WWPN word 0 and word 1 for adapter number 1:

```
oslpcfg scriptvwwpn n=1 w0=20A2D6B8 w1=C920A1B2
```

4.3.8.7 Creating a Log (logfile)

The `logfile` command creates a log file with a specified directory and file name. The default log file is `lpcfglog.txt` and is created in the system's current directory.

NOTE: This command is supported only in script files. Make it the first command in the script.

To create a log file, type the following command in the script file:

```
logfile l=<filename>
```

The following example creates a file called `lplog.txt` in the `d:\log` directory:

```
logfile l=d:\log\lplog.txt
```

Results of all commands are recorded in a log file. Unless otherwise specified by the logfile command, the default log file is `lpcfglog.txt` in the current directory.

Chapter 5: ESXi 8.0 Functionality

This section provides the Elxflash and LpCfg utility functionality for ESXi 8.0.

To use multiple firmware files and retain firmware files after reboot, create the firmware subdirectory in the datastore and use the -d switch. The datastore has more memory, and its contents are retained after reboot.

5.1 Elxflash Utility

In ESXi 8.0, the Elxflash Stand-alone utility has been changed to comply with VMware security requirements. To install the ESXi 8.0 utility, perform the following steps:

1. On the ESXi host, install the component as follows:
   ```
   esxcli software component apply -d /<pathname>/Broadcom-ELX-elxflash-<component_version>-1OEM-800.1.0.zip
   ```
   where `<component_version>` represents the Emulex component version.
2. Reboot the host.

   After the host is rebooted, the `elxflash` namespace appears under `esxcli`.

The root installation directory is `/opt/elxflash`.

NOTE: In ESXi 8.0, the `elxflash` commands are replaced by `esxcli elxflash` commands.

NOTE: In the ESXi 8.0 download commands, the `<mode>` parameter must be specified. The `<mode>` parameter is either `auto` or `manual`.

NOTE: The following Elxflash commands are not supported in the ESXi 8.0 operating system:

- Log (`/log`); Elxflash logs are written to the `/var/log/elxflash.log` file
- Silent (`/s`)
- Verbose (`/v`)
- XML output (`/xml`)

5.1.1 Using the fwmatrix.txt File Mode

To use the `fwmatrix.txt` file, you must create or update the firmware subdirectory on datastore with the.appropriate firmware images. The `fwmatrix.txt` file includes adapter entries followed by a firmware image file entry.

NOTE: To avoid a firmware update failure, ensure that there are no spaces in the model names in the `fwmatrix.txt` file. Copy the `fwmatrix.txt` file in the same location as of the firmware folder on datastore. Not inside the firmware folder.

When the Elxflash utility uses the `fwmatrix.txt` file, it locates the matching adapter model entry and then attempts to download the specified firmware image file. If there are multiple entries, the utility uses only the last matching entry for the adapter that is being evaluated.

Specify the use of the `fwmatrix.txt` file by using the `esxcli elxflash manual <command> [command_option] -d <absolute_path>` command

5.1.2 Using the Autodiscovery Mode

If the `auto` mode is used with an operational switch, such as `update`, the Elxflash utility automatically discovers adapters. Using the `firmware` subdirectory, the utility performs the operation specified by the switch on each adapter.

5.1.3 Autodiscovery (auto)

Usage: `esxcli elxflash auto`

The `auto` switch instructs the Elxflash utility to ignore the `fwmatrix.txt` file, to automatically discover local adapters, and to perform specified operations employing an additional switch using the `firmware` subdirectory.

The `auto` switch must be used with an additional operational switch, such as one of the following switches:

- `downgrade`
- `forcefirmware`
- `rewrite`
- `update`

Example usage:

`esxcli elxflash auto update` – Updates the firmware using the `firmware` subdirectory.

- The `fwmatrix.txt` file is ignored. The desired versions of firmware must be in the `firmware` subdirectory.
- Using the `firmware` subdirectory, the Elxflash utility automatically discovers the best matching firmware for each installed and supported adapter. See Section 5.1.8, Image Version Display (imageversion -i <image_file>) for more information about the `firmware` subdirectory.
- If multiple versions of firmware are found for an adapter, the Elxflash utility uses the most recent version when performing the firmware update.

5.1.4 Downgrade (downgrade)

Usage: `esxcli elxflash <mode> downgrade`

The `downgrade` switch downgrades the firmware of each adapter if the currently installed versions are more recent than the downgrade versions.

Example usage:

`esxcli elxflash auto downgrade` – Downgrades the firmware using autodiscovery.

- The `fwmatrix.txt` file is ignored. The desired downgrade versions of firmware must be in the `firmware` subdirectory. See Section 5.1.8, Image Version Display (imageversion -i <image_file>) for more information about the `firmware` subdirectory.
- If the downgrade versions are older than the currently installed versions on the adapter, the downgrade versions are downloaded to the adapter.
- If multiple downgrade versions of firmware are found for an adapter, the next-previous downgrade versions are downloaded to the adapter.

`esxcli elxflash manual downgrade` – Downgrades the firmware using the `fwmatrix`.txt file.

- For each installed and supported adapter, the current firmware version is compared to the version specified in `fwmatrix.txt`.
- If the downgrade version in `fwmatrix.txt` is older than the currently installed version, the downgrade version of firmware is downloaded to that adapter.

5.1.5 Force Firmware (forcefirmware)

Usage `esxcli elxflash <mode> forcefirmware`

The `forcefirmware` switch forces a firmware download to an adapter regardless of the current version on the adapter, and it is performed regardless of any additional operational switches given on the command line.

Example usage:

`esxcli elxflash auto forcefirmware` – Force a firmware download using autodiscovery.

- The `fwmatrix.txt` file is ignored. The desired versions of firmware must be in the `firmware` subdirectory. See Section 5.1.8, Image Version Display (imageversion -i <image_file>) for more information about the `firmware` subdirectory.
- If multiple versions of firmware are found for an adapter, the Elxflash utility uses the most recent version when performing the firmware download.

`esxcli elxflash manual forcefirmware` – Forces a firmware download using the `fwmatrix.txt` file.

- For each installed and supported adapter, this command forces a download of firmware using the versions specified in the `fwmatrix.txt` file.

5.1.6 Default Firmware Directory (-d <absolute path>)

Usage: `-d <absolute_path>` or `--dir=<absolute_path>`

The `-d` switch specifies the absolute path of the `firmware.txt` file or the `firmware` folder, depending on the `<mode>` parameter.

- For the `esxcli elxflash manual <options>` command, the default `firmware` subdirectory is `/opt/elxflash/vital`. Create a `firmware` subdirectory under this directory and copy the firmware images specified in the `fwmatrix.txt` file into it.
- For the `esxcli elxflash auto <options>` command, the default firmware subdirectory is `/opt/elxflash/vital`. Create a `firmware` subdirectory under this folder and copy the firmware images you want to use into it.

NOTE: Do not include `/firmware` at the end of `<absolute_path>`.

NOTE: The `/opt/elxflash/vital` directory has a memory limit of 8 MB, and the `firmware` subdirectory is deleted when the system is rebooted. To use multiple firmware files and retain firmware files after reboot, create the `firmware` subdirectory in the datastore and use the `-d` switch. The datastore has more memory, and its contents are retained after reboot.

Example usage:

`esxcli elxflash auto forcefirmware -d /opt/elxflash/vital/firmware` – Forces a firmware download using autodiscovery.

- Elxflash looks for the `firmware` subdirectory in the `/opt/elxflash/vital` directory.
- The `fwmatrix.txt` file is ignored.
- Each firmware image file must be placed in the `/opt/elxflash/vital/firmware` subdirectory.
- If multiple versions of firmware are found for an adapter, Elxflash uses the most recent versions when performing the firmware download.

`esxcli elxflash manual forcefirmware -d /opt/elxflash/vital/firmware` – Forces a firmware download using the `fwmatrix.txt` file.

- For each installed and supported adapter, a forced download of firmware occurs using the versions specified in the `fwmatrix.txt` file.

5.1.7 Help

Usage: `esxcli elxflash` or `esxcli elxflash <`*`mode`*`> <`*`command`*`> [command_option] --help`

The `help` switch displays a help message with instructions for using the Elxflash utility.

5.1.8 Image Version Display (imageversion -i <image_file>)

Usage: `esxcli elxflash imageversion -i <`*`image_file`*`>`)

The `imageversion` command shows the firmware image file's version number. You must specify a path to the firmware image file for the command to decode the image file's version.

Example usage:

```
esxcli elxflash imageversion -i firmware/prism_A14.2.123.45
```

The following information is returned:

```
14.2.123.45
```

5.1.9 Preview (-p 1)

Usage: `-p 1` or `--preview=1`

The preview switch provides a download preview of all adapters the Elxflash utility can update using either autodiscovery or the `fwmatrix.txt file`. The preview switch can be used with any of the operational switches, such as the following:

- `downgrade`
- `forcefirmware`
- `rewrite`
- `update`

When the preview switch is used, the Elxflash utility displays a download summary, but it does not actually perform the download.

NOTE: If the preview switch is used, the `Status=<`*`description`*`>` field displays `Preview`.

Each adapter's download preview displays the adapter's old and new image versions. The old image version represents the image version that is currently on the adapter. The new image version represents the image version the Elxflash utility would use during a download.

Example usage:

`esxcli elxflash auto update -p 1` – Previews an upgrade of firmware using autodiscovery.

- The `fwmatrix.txt` file is ignored. The desired update versions of firmware must be in the `firmware` subdirectory.
- If the update versions are later than the currently installed versions on the adapter, the Elxflash utility provides a download preview for each adapter that can be updated.
- If multiple update versions of firmware are found for an adapter, the Elxflash utility provides a download preview using the most recent versions.

Example usage:

`esxcli elxflash manual update -p 1` – Previews an upgrade of firmware using the `fwmatrix.txt` file.

- For each installed and supported adapter, the current firmware versions are compared to the versions specified in the `fwmatrix.txt` file.
- If the update versions in the `fwmatrix.txt` file are more recent than the currently installed versions, the Elxflash utility provides a download preview of firmware for each adapter that can be updated.

5.1.10 Query (-q 1)

Usage: `-q 1` or `--query=1`

The query switch displays an adapter's model, WWN, PCI_ID, firmware version, and flash part type. The query switch can be used with any switch.

Example usage:

```
esxcli elxflash -q 1
```

The following information is returned:

```
HBA=<model>, Port Type=FC, WWN =<wwn>, Firmware=<version>, Flash Part=<type>
```

5.1.11 Rewrite (rewrite)

Usage: `esxcli elxflash <mode> rewrite`

The `rewrite` switch updates the firmware of each adapter if the installed versions are earlier than, or the same as, the rewrite versions.

Example usage:

`esxcli elxflash auto rewrite` – Rewrites the firmware using autodiscovery.

- The `fwmatrix.txt` file is ignored. The desired rewrite versions of firmware must be in the `firmware` subdirectory.
- If the rewrite versions are more recent than, or the same as, the versions installed on the adapter, the rewrite versions are downloaded to the adapter.
- If multiple rewrite versions of firmware are found for an adapter, the most recent versions are downloaded to the adapter.

`esxcli elxflash manual rewrite` – Rewrites the firmware using the `fwmatrix.txt` file.

- For each installed and supported adapter, the current firmware version is compared to the version in the `fwmatrix.txt` file.
- If the currently installed versions are less than or equal to the rewrite versions in the `fwmatrix.txt` file, the rewrite versions of firmware are downloaded to the adapter.

5.1.12 Update (update)

Usage: `esxcli elxflash <mode> update`

The `update` switch updates the firmware of each adapter if the currently installed versions are older than the update versions.

Example usage:

`esxcli elxflash auto update` – Upgrades the firmware using autodiscovery.

- The `fwmatrix.txt` file is ignored. The desired update versions of firmware must be in the `firmware` subdirectory.
- If the update versions are newer than the currently installed versions on the adapter, the update versions are downloaded to the adapter.
- If multiple update versions of firmware are found for an adapter, the most recent version is downloaded to the adapter.

`esxcli elxflash manual update` – Upgrades the firmware using the `fwmatrix.txt` file.

- For each installed and supported adapter, the current firmware version is compared to the version specified in `fwmatrix.txt`.
- If the update versions in `fwmatrix.txt` are more recent than the currently installed version, the update versions of firmware are downloaded to that adapter.

5.1.13 VPD (-m vpd)

Usage: `-m vpd`

The `-m vpd` command displays the VPD model name for supported adapters. You must use the `-m vpd` command with one of the following switches:

- `downgrade`
- `-q 1` or `--query=1`

NOTE: The `-m vpd` command does not display any information if it is used without one of the preceding switches. If you attempt to use the `-m vpd` command without one of the preceding switches, an error message is displayed.

Example usage:

```
esxcli elxflash manual downgrade -m vpd
```

- If the `-m vpd` command is used with the `fwmatrix.txt` file, the model name in the `fwmatrix.txt` file must match the VPD model name reported by Elxflash. Supported adapters include legacy FC adapters that support VPD.

```
esxcli elxflash -m vpd
```

5.2 LpCfg Utility

In ESXi 8.0, the LpCfg utility has been changed to comply with VMware security requirements.

The `linlpcfg` utility is installed with the elxflash component. To run `linlpcfg` utility, type the following command:

```
esxcli linlpcfg <valid command> [option1] ...[optionn]
```

5.2.1 Supported Commands

Use standard `esxcli` commands to run the LpCfg utility.

The following LpCfg commands are supported in the ESXi 8.0 operating system:

- `hba`
- `loopbacktest`
- `bootparams`
- `config`
- `factorydefaults`
- `flashpart`
- `fwparams`
- `hbaattr`
- `listboot`
- `listrev`
- `nvwwn`
- `script`
- `version`
- `vpd`
- `wwn`
- `xcvrdata`

5.2.2 Help Command and General Adapter Management Commands

Use the following commands to view the online help, reset the adapter, or reset the adapter to factory defaults.

5.2.2.1 Viewing the Syntax for Commands (help)

To view the syntax for all available commands, type the following command:

```
esxcli linlpcfg --help
```

To view the syntax for a specific command, type the following command:

```
esxcli linlpcfg <command> [command_option] --help
```

For example, the following command:

```
esxcli linlpcfg hba firmware download --help
```

returns a response similar to the following:

```
download n=<adapter> i=<path/image_filename>
```

5.2.2.2 Resetting an Adapter (reset)

This command resets a specific adapter or all adapters in the system.

ATTENTION: Do not execute a reset on an FC adapter that is being used to boot from SAN. The reset might cause a loss of connectivity to the SAN and possible loss of data. To reset an FC adapter, make sure the adapter is not currently being used to boot from SAN. There are two ways to perform this action:

- Move the adapter you want to reset to a non-boot-from-SAN host, and reset it from there.
- If the host with the reset target adapter is also hosting other boot-from-SAN adapters, carry out a boot from SAN using one of the other boot-from-SAN adapters. The intended target adapter can now be reset because it is not being actively used for boot from SAN.

To reset one adapter, type the following command:

```
esxcli linlpcfg hba reset n=<adapter_number>
```

To reset all adapters in the system, type one following command:

```
esxcli linlpcfg hba reset n=<all>
```

5.2.2.3 Resetting an Adapter to the Factory Defaults (factorydefaults)

This command returns the adapter to the factory default profile and configuration. A reboot is required for the changes to take effect.

NOTE: After a successful factory reset the adapter is offline. Perform an immediate reboot to complete the reset and return the adapter to full functionality.

ATTENTION: Do not restore the factory defaults on an FC adapter that is being used to boot from SAN. The `factorydefaults` command might cause a loss of connectivity to the SAN and possible loss of data. To restore the factory defaults on an FC adapter, make sure the adapter is not currently being used to boot from SAN. There are two ways to perform this action:

- Move the adapter on which you want to restore the defaults to a non-boot-from-SAN host, and perform the restore defaults command from there.
- If the host with the restore defaults target adapter is also hosting other boot-from-SAN adapters, carry out a boot from SAN using one of the other boot-from-SAN adapters. The factory defaults can now be restored on the intended target adapter because it is not being actively used for boot from SAN.

To return an adapter to its default settings, type the following command:

```
esxcli linlpcfg factorydefaults set n=<adapter_number|all>
```

The following example returns adapter number 1 to its default settings:

```
esxcli linlpcfg factorydefaults set n=1
```

5.2.3 Viewing Adapter Information

Use the following commands to view different kinds of adapter information, such as the VPD model name, boot device information, and adapter attributes.

5.2.3.1 Viewing the LpCfg Utility Version Information (version)

This command shows the LpCfg utility version information.

To view this information, type the following command:

```
esxcli linlpcfg version get
```

5.2.3.2 Viewing the VPD (vpd)

This command shows the VPD of the adapter specified by its number.

To display the VPD, type the following command:

```
esxcli linlpcfg vpd get n=<adapter_number>
```

5.2.3.3 Viewing Adapter Attributes (hbaattr)

This command displays adapter information.

To list the adapter attributes for all installed adapters, type the following command:

```
esxcli linlpcfg hbaattr get
```

To list the adapter attributes for one adapter, type the following command:

```
esxcli linlpcfg hbaattr get n=<adapter_number>
```

NOTE: The `hbaattr` command output includes Firmware Status information. If a reboot is required to activate new firmware, Firmware Status displays the following text:

```
Reboot the system to activate new firmware.
```

If an optional reboot is required to enable a new feature, Firmware Status displays the following text:

```
The new firmware is activated. Some features require an optional reboot. Refer to the
Adapter's Firmware and Boot Code Release Notes for details.
```

For a list of features that require a reboot to be enabled, refer to the firmware and boot code release notes for the adapter.

5.2.3.4 Viewing All Adapters in the System (listhba)

This command lists all installed adapters in the system. Information includes the adapter number, the IEEE address assigned by the manufacturer, the functional firmware, the adapter type, and possible mailbox errors.

To list all adapters in the system, type the following command:

```
esxcli linlpcfg hba list
```

5.2.3.5 Viewing the WWN of All Adapters in the System (listwwn)

This command lists all adapters installed in the system and shows the factory-assigned WWN, the nonvolatile WWPN, and the WWNN used to identify an adapter in the SAN.

The factory-assigned WWN is an IEEE address that cannot be changed in the field. The nonvolatile WWN can be modified in the field and persists after a restart of the operating system. The full factory-assigned WWN and nonvolatile WWN are a concatenation of the two 8-character values (word 0 and word 1) that are shown for each. You can modify the nonvolatile WWPN and WWNN using the `scriptwwpn` and `scriptwwnn` commands.

If the system does not have any Emulex adapters installed, error code 45 is returned.

To show the WWN information, type the following command:

```
esxcli linlpcfg wwn list
```

5.2.3.6 Viewing Firmware Program Revisions (listrev)

This command shows the firmware versions in the adapter's flash memory, specified by their numbers.

To show revisions, type the following command:

```
esxcli linlpcfg listrev get n=<adapter_number>
```

The following example lists the firmware revision information for adapter number 3:

```
esxcli linlpcfg listrev get n=3
```

5.2.3.7 Viewing Selected Configuration Regions (readconfig)

This command shows the contents of the selected configuration region up to the initialized length or the specified byte count (if the initialized length is less than the specified byte count). The following region numbers are valid:

- 0
- 8
- 9
- 10
- 14
- 16 (which is remapped to 32)
- 32

You must initialize the configuration region first by writing data to it.

To read a configuration, type the following command:

```
esxcli linlpcfg config read n=<adapter_number> r=<region_number> l=<byte_count>
```

The following example reads the configuration for adapter number 1, region 0, byte count 20:

```
esxcli linlpcfg config read n=1 r=0 l=20
```

5.2.3.8 Viewing the Flash Device (getflashpart)

This command displays the flash device that is installed on the adapter. To view the flash device information, type the following command:

```
esxcli linlpcfg flashpart get n=<adapter_number>
```

The following example displays the flash device information for adapter number 1:

```
esxcli linlpcfg flashpart get n=1
```

5.2.4 Firmware Commands

5.2.4.1 Viewing Firmware Parameters (getfwparams)

This command displays the topology parameter.

To view the topology parameter, type the following command:

```
esxcli linlpcfg fwparams get n=<adapter_number>
```

5.2.4.2 Setting Firmware Parameters (setfwparam)

This command configures the topology parameter. To set the topology, type the following command:

```
esxcli linlpcfg fwparams set n=<adapter_number> topology <value>
```

where `<value>` is one of the following values:

- `0`=Auto (loop first)
- `1`=Auto (point-to-point first)
- `2`=Loop
- `3`=Point-to-point

NOTE: Point-to-point and FC-AL (loop) topologies are supported on the LPe31000-series and LPe32000-series adapters, and on the LPe35000-series and LPe36000-series adapters at speeds up to and including 16Gb/s. Point-to-point is the only topology supported on the LPe35000-series and LPe36000-series adapters at speeds greater than 16Gb/s.

After the `setfwparam` command is issued, you must reset the adapter port to activate the new firmware parameter setting.

5.2.5 Firmware Download Commands

The following firmware download commands include a command to download a firmware file and a command to access the flash device directly.

5.2.5.1 Downloading a File (download)

This command downloads a firmware file to a specific adapter.

The adapter name is the name that appears when you run the `listhba` command. For more information on the `listhba` command, see Section 5.2.3.4, Viewing All Adapters in the System (listhba).

To download a firmware image file to an adapter specified by its number, type the following command:

```
esxcli linlpcfg hba firmware download n=<adapter_number> i=<firmware_image_filename>
```

The following example downloads the `prism_A14.0.123.45.grp` firmware to adapter number 6; in this example, the firmware file is for an LPe35000 adapter:

```
esxcli linlpcfg hba firmware download n=6 i=prism_A14.2.123.45
```

To verify the reset or boot status of new firmware, issue the `hbaatrib` command and locate the Firmware Status information. See Section 5.2.3.3, Viewing Adapter Attributes (hbaattr), for more information.

In some cases, a firmware update requires either a firmware reset or a basic PCIe reset, depending on the features available in the new firmware. A firmware reset is performed automatically if it is needed, regardless of the operating system.

If a firmware reset or basic PCIe reset occurs when the firmware is downloaded, a message similar to the following appears:

```
Download and pci reset successfully completed
```

In some cases, a full reboot is required to activate new firmware or to enable a new feature. In that case, a message similar to one of the following messages appears after the firmware download is complete:

```
Download successfully completed. Please reboot the system to activate new firmware.
Download completed. Some features require an optional reboot. Refer to the Adapter's Firmware and Boot
Code Release Notes for details.
```

For a list of features that require a reboot to be enabled, refer to the firmware and boot code release notes for the adapter.

5.2.6 World Wide Name Commands

The following commands allow you to save WWN data to a file, and restore WWN data while updating the NVPARMS. You can also restore the NVPARMS and the IEEE address.

5.2.6.1 Saving WWN Data to a File (savewwn)

This command reads the original words 0 and 1 of the IEEE address, installed by manufacturing, from configuration regions 16 (or 32) of the adapter (specified by its number), and it saves the configuration region information in the selected WWN file.

NOTE: The output of this command is a binary file. Use the `hexdump` utility to view its contents.

To save the WWN data to a file, type the following command:

```
esxcli linlpcfg wwn save n=<adapter_number> c=<wwn_filename>
```

The following example reads the configuration region information on adapter number 4 and saves it to the contents of the `ctwwn.sav` file:

```
esxcli linlpcfg wwn save n=4 c=ctwwn.sav
```

5.2.6.2 Restoring the WWN and Updating the NVPARMS (restorewwn)

This command restores words 0 and 1 of the IEEE address from a specified file created with the `savewwn` command and uses them to update the NVPARMS port name with this IEEE address.

ATTENTION: Do not run a `restorewwn` command on an FC adapter that is being used to boot from SAN. The `restorewwn` command might cause a loss of connectivity to the SAN and possible loss of data. To restore the WWN and update NVPARMS on an FC adapter, make sure the adapter is not currently being used to boot from SAN. There are two ways to perform this action:

- Move the adapter on which you want to use the `restorewwn` command to a non-boot-from-SAN host, and run `restorewwn` from there.
- If the host with the `restorewwn` target adapter is also hosting other boot-from-SAN adapters, carry out a boot from SAN using one of the other boot-from-SAN adapters. You can now restore the WWN and update the NVPARMS on the intended target adapter because it is not being actively used for boot from SAN.

To restore the WWN, type the following command:

```
esxcli linlpcfg wwn restore n=<adapter_number> c=<wwn_filename>
```

The following example updates the NVPARMS on adapter number 4 with the `ctwwn.sav` file.

```
esxcli linlpcfg wwn restore n=4 c=ctwwn.sav
```

5.2.6.3 Restoring the NVPARMS (restorenvwwn)

ATTENTION: Do not run a `restorenvwwn` command on an FC adapter that is being used to boot from SAN. The `restorenvwwn` command might cause a loss of connectivity to the SAN and possible loss of data. To restore the NVPARMS on an FC adapter, make sure the adapter is not currently being used to boot from SAN. There are two ways to perform this action:

- Move the adapter on which you want to use the restorenvwwn command to a non-boot-from-SAN host, and run the command from there.
- If the host with the `restorenvwwn` target adapter is also hosting other boot-from-SAN adapters, carry out a boot from SAN using one of the other boot-from-SAN adapters. You can now restore the NVPARMS on the intended target adapter because it is not being actively used for boot from SAN.

This command restores the nonvolatile WWPN and WWNN to the adapter, replacing any volatile WWPN and WWNN data, without powering off the adapter.

If the adapter does not have firmware that supports the volatile WWN, the following error message appears:

```
Write Volatile Parms Error. Reported Error 48
```

If this error occurs, install firmware that supports the volatile WWN.

To restore the nonvolatile WWN, type the following command:

```
esxcli linlpcfg nvwwn restore n=<adapter_number>
```

The following example restores the nonvolatile WWN on adapter number 2.

```
esxcli linlpcfg nvwwn restore n=2
```

5.2.6.4 Restoring the IEEE Address (restoredefwwn)

This command reads the IEEE address (assigned by the manufacturer) and writes it to the nonvolatile WWPN and WWNN.

ATTENTION: Do not run a `restoredefwwn` command on an FC adapter that is being used to boot from SAN. The `restoredefwwn` command might cause a loss of connectivity to the SAN and possible loss of data. To restore the IEEE address on an FC adapter, make sure the adapter is not currently being used to boot from SAN. There are two ways to perform this action:

- Move the adapter on which you want to use the `restoredefwwn` command to a non-boot-from-SAN host, and run the command from there.
- If the host with the `restoredefwwn` target adapter is also hosting other boot-from-SAN adapters, carry out a boot from SAN using one of the other boot-from-SAN adapters. You can now restore the IEEE address on the intended target adapter because it is not being actively used for boot from SAN.

To restore the IEEE address, type the following command:

```
esxcli linlpcfg nvwwn restoredef n=<adapter_number>
```

The following example restores the IEEE address on adapter number 2.

```
esxcli linlpcfg nvwwn restoredef n=2
```

5.2.6.5 Boot Code Commands

NOTE: The parameters and keywords for boot commands are the same in the ESXi 8.0 operating system as they are in other operating systems.

To set the boot device with the offline utility, run the following commands in this order:

1. Use the `listboot` command to verify that the boot code is present. See Section , esxcli linlpcfg vpd get n=<*adapter_number*>, for more information.

   Example:

   ```
   esxcli linlpcfg listboot get n=1
   ```

   Sample return for an HBA with x86 BootBIOS enabled:

   ```
   Boot Code (enabled) = 14.2.123.4
   fc_universal_bios_version = 14.2.234.5
   fc_x86_bios_version = 14.2.345.6
   fc_efi_bios_version = 14.2.567.8
   Command completed, NO Error
   ```

   Sample return for an HBA with x86 BootBIOS disabled:

   ```
   Boot Code (disabled) = 14.2.123.4
   fc_universal_bios_version = 14.2.234.5
   fc_x86_bios_version = 14.2.345.6
   fc_efi_bios_version = 14.2.567.8
   Command completed, NO Error
   ```

2. Use the `setbootparam` command to perform the following tasks:
   - Enable boot code (`enableadapterboot` parameter).
   - Configure the HBA to boot from SAN (`enablebootfromsan` parameter).
   - Set alternate boot devices (`d_id`, `lun`, `targetwwpn`, or `targetid` parameter).
3. Run the `esxcli linlpcfg bootparams get` command (Section 5.2.6.7, Getting Boot Parameters (getbootparams)) to verify the configuration.
4. Configure the system BIOS so the adapter boot device is the highest in the boot order.

5.2.6.6 Configuring Boot Parameters (setbootparam)

This command changes the adapter parameters and the boot device parameters. You can change adapter parameters and boot device parameters for x86 boot and EFI boot.

NOTE: If you are changing an adapter parameter, omit the `bootdev` keyword and value; otherwise, an error is reported.

If you are changing a boot device parameter for x86 boot and EFI boot, you must provide the `bootdev` keyword and value.

Changes made to x86 boot settings are also applied to EFI boot settings, and changes made to EFI boot settings are also applied to x86 boot settings.

To configure an adapter parameter, type the following command:

```
esxcli linlpcfg bootparams set n=<adapter_number> <type> <param> <param_value>
```

To configure a boot device parameter, type the following command:

```
esxcli linlpcfg bootparams set n=<adapter_number> <type> <param> <param_value> bootdev <bootdev_value>
```

Parameters

`adapter_number`	The number of the adapter to be configured.
`type`	The supported boot type (provided in Table 4, Boot Parameter): `x86`: x86Boot `efi`: UEFIBoot
`param`	See Table 4, Boot Parameter.
`param_value`	See Table 4, Boot Parameter.
`bootdev_value`	Boot device entry number (`0` through `7`).

NOTE: OpenBoot is no longer supported. The feature will be disabled in a future release.

The following table provides the available boot parameters, their `param_value` values, and their supported boot types.

Table 4: Boot Parameter

Parameter Type	Parameter Name	Description	Supported Boot Types	param_value
Adapter	`autoscan`	The `autoscan` parameter allows you to set auto scan and enable the first device in the boot entry list to issue a Name Server Inquiry. The `autoscan` parameter is available only if none of the eight boot entries are configured to boot from DID or WWPN. The factory default is `0`. If more than one adapter with the same PCI bus number exists in the system, and each has a boot drive attached, the first PCI-scanned adapter is the boot adapter.	`x86`	`0`=Disabled `1`=First device `2`=First LUN 0 device `3`=First non-LUN 0 device
Adapter	`autobootsectorenable`	The `autobootsectorenable` parameter automatically defines the boot sector of the target disk for the migration boot process. This applies only to HP MSA1000 arrays. If no partition is on the target, the default boot sector format is `63` sectors. The default setting is `0`.	`x86`	`0`=Disable `1`=Enable

Table 4: Boot Parameter (Continued)

Parameter Type	Parameter Name	Description	Supported Boot Types	param_value
Adapter	`boottargetscan`	The `boottargetscan` parameter allows you to specify the method of scanning for boot targets. The following methods can be selected: ■ Boot path from NVRAM targets: A boot scan discovers only LUNs that are saved to the adapter's NVRAM. Select up to eight attached devices to use as potential boot devices. Limiting discovery to a set of eight selected targets can greatly reduce the time it takes for the EFIBoot driver to complete discovery. ■ Boot path from discovered targets: A boot scan discovers all devices that are attached to the FC port. Discovery can take a long time on large SANs if this option is selected. ■ Do not create a boot path. ■ EFI FC scan level: Setting an EFI FC scan level allows third-party software to toggle between a boot path from NVRAM and a boot path from discovered targets by manipulating a UEFI system NVRAM variable. ■ After the scan is set to an EFI FC scan level, the scan method can be changed without entering the UEFI boot configuration utility. You can establish the initial setting as either NVRAM targets (`3`) or discovered targets (`4`). ■ If you select one of the EFI FC scan levels settings, the scan is determined by the value of the `EFIFCScanLevel` variable maintained by the UEFI system firmware and boot code. The value of this variable can be changed either by using the `bootargetscan` parameter or by using third-party software.	`efi`	`0`=Boot path from NVRAM targets `1`=Boot path from discovered targets `2`=Do not create boot path `3`=EFI FC scan level: NVRAM targets `4`=EFI FC scan level: Discovered targets
Adapter	`edd30enable`	EDD 3.0 provides additional data to the operating system bootloader during the INT-13h function 48h (get device parameters) call. This information includes the path to the boot device and disk size. The default setting for EDD 3.0 is `1`. **NOTE:** If EDD 3.0 is disabled, EDD 2.1 is enabled.	`x86`	`0`=Disable `1`=Enable

Table 4: Boot Parameter (Continued)

Parameter Type	Parameter Name	Description	Supported Boot Types	param_value
Adapter	enableadapterboot	The enableadapterboot parameter enables or disables the boot code for the specified adapter. If the boot code is disabled, the adapter does not boot from the SAN, regardless of the value of the enablebootfromsan parameter. If the boot code is enabled, the adapter boots from the SAN if the enablebootfromsan parameter is also enabled.	x86 and efi	0=Disable 1=Enable
Adapter	enablebootfromsan	The enablebootfromsan parameter enables the adapter to boot from SAN. **NOTE:** The enableadadapterboot parameter must also be set to 1 for an adapter to boot from SAN. **NOTE:** To boot from SAN using NVMe, you must enable both the enablebootfromsan parameter and the enablenvmebootfromsan parameter.	x86 and efi	1=Disable 1=Enable
Adapter	enablefabl	FABL is no longer supported. The feature will be disabled in a future release.		
Adapter	envvarenable	The envvarenable parameter sets the boot controller order if the system supports the environment variable. The default setting is 0.	x86	0=Disable 1=Enable
Adapter	maxlunspertarget	The maxlunspertarget sets the maximum number of LUNs that are polled during device discovery.	efi	1 through 4096
Adapter	plogiretrytimer	The plogiretrytimer parameter is useful for Tachyon-based RAID arrays. In rare situations, a Tachyon-based RAID array resets itself and the port goes offline temporarily. When the port returns to operation, the PLOGI scans the loop to discover this device. The PLOGI retry interval is the time it takes for one PLOGI to scan the whole loop (if 126 AL_PAs are on the loop). Changes made to x86 are also applied to efi. Changes made to efi are also applied to x86.	x86 and efi	0=None 1=50 ms 2=100 ms 3=200 ms

Table 4: Boot Parameter (Continued)

Parameter Type	Parameter Name	Description	Supported Boot Types	param_value
Adapter	spinupdelayenable	The spinupdelayenable parameter allows you to enable or disable the disk spin-up delay. The factory default setting is 0. If at least one boot device has been defined, and the spin-up delay is enabled, the BIOS searches for the first available boot device. ■ If a boot device is present, the BIOS boots from it immediately. ■ If a boot device is not ready, the BIOS waits for the spin-up delay and, for up to 3 additional minutes, continues the boot scanning algorithm to find another multi-boot device. If boot devices have not been defined and the autoscan parameter is enabled, the BIOS waits for 5 minutes before scanning for devices. ■ In a private loop, the BIOS attempts to boot from the lowest target AL_PA it finds. ■ In an attached fabric, the BIOS attempts to boot from the first target found in the NameServer data.	x86	0=Disable 1=Enable
Adapter	startunitcommandenable	The startunitcommandenable parameter allows a start-unit command to be sent to a particular disk. This option might be needed if the disk you want to boot from is not yet spun up or started. You must know the specific LUN to which to issue the SCSI start unit command. The default setting is 0.	x86	0=Disable 1=Enable
Boot device	d_id	Specify the boot device by its device ID.	x86	Use the D_ID of the boot device (in decimal)
Boot device	lun	Specify the boot device by its starting LUN. Changes made to x86 are also applied to efi. Changes made to efi are also applied to x86.	x86 and efi	0 through 255
Boot device	targetwwpn	Specify the boot device by its target WWPN. Changes made to x86 are also applied to efi. Changes made to efi are also applied to x86.	x86 and efi	Use the target WWPN in the following format: nn:nn:nn:nn:nn:nn:nn:nn

Table 4: Boot Parameter (Continued)

Parameter Type	Parameter Name	Description	Supported Boot Types	param_value
NVMe adapter	enablenvmebootfromsan	Enable BFS on NVMe over FC. **NOTE:** To boot from SAN using NVMe, you must enable both the enablebootfromsan parameter and the enablenvmebootfromsan parameter.	efi	0=Disable 1=Enable
NVMe boot device	namespaceid	Specify the boot device by its NVMe namespace ID value.	efi	0 through 0xFFFFFFFF
NVMe boot device	nvmewwpn	Specify the boot device by its NVMe WWPN.	efi	Use the NVMe WWPN in the following format: nn:nn:nn:nn:nn:nn:nn:nn
NVMe boot device	nvmewwnn	Specify the boot device by its NVMe WWNN.	efi	Use the NVMe WWNN in the following format: nn:nn:nn:nn:nn:nn:nn:nn
NVMe boot device	subsystemnqn	Specify the boot device by its subsystem NQN.	efi	Use the subsystem NQN string, which must conform to NQN standards. The maximum length is 233 characters.

5.2.6.7 Getting Boot Parameters (getbootparams)

This command displays the configured adapter parameters and boot device parameters.

To display boot parameters, type the following command:

```
esxcli linlpcfg bootparams get n=<adapter_number> <boot_type>
```

where <*boot_type*> is one of the following values:

- x86 Boot BIOS
- UEFI Boot

NOTE: OpenBoot is no longer supported. The feature will be disabled in a future release.

NOTE: All Emulex adapters do not support the same boot parameters. Boot parameters that are not supported by the adapter are not displayed.

Following is an example of the boot parameters for the x86 boot type on an LPe35000-series HBA:

```
esxcli linlpcfg bootparams get n=1 x86
Command: bootparams get n=1 x86
BootParams Get: Success
Parameter Name         Value
--------------         -----
 Standard Parameters
  EnableAdapterBoot        1   Enabled
  EnableBootFromSAN        1   Enabled
 Advanced Parameters
  AutoBootSectorEnable     0   Disabled
  AutoScan                 0   Disabled
  EDD30Enable              1   Enabled
  EnvVarEnable             0   Disabled
  PlogiRetryTimer          0   None
  SpinupDelayEnable        0   Disabled
  StartUnitCommandEnable   0   Disabled
 Boot Device Parameters
  Entry  Lun    D_ID        TargetWwpn
  -----  ---  ------  -----------------------
      0  000  000000  00:00:00:00:00:00:00:00
      1  000  000000  00:00:00:00:00:00:00:00
      2  000  000000  00:00:00:00:00:00:00:00
      3  000  000000  00:00:00:00:00:00:00:00
      4  000  000000  00:00:00:00:00:00:00:00
      5  000  000000  00:00:00:00:00:00:00:00
      6  000  000000  00:00:00:00:00:00:00:00
      7  000  000000  00:00:00:00:00:00:00:00
```

5.2.7 Diagnostic Tests

The following commands let you run diagnostic tests on your adapter.

5.2.7.1 Running the External Loopback Test (extloopback)

ATTENTION: Do not perform an `extloopback` test on an FC adapter that is being used to boot from SAN. The `extloopback` command might cause a loss of connectivity to the SAN and possible loss of data. To perform an `extloopback` command on an FC adapter, make sure the adapter is not currently being used to boot from SAN. There are two ways to perform this action:

- Move the adapter on which you want to perform the external loopback test onto a non-boot-from-SAN host, and perform the test from there.
- If the host with the test target adapter is also hosting other boot-from-SAN adapters, carry out a boot from SAN using one of the other boot-from-SAN adapters. The `extloopback` command can now be run on the intended target adapter because it is not being actively used for boot from SAN.

This command runs the external loopback test. You must put a loopback plug in each adapter port to be tested. You can test a specific adapter in the system. Specify the number of times you want the test to repeat, and direct the test response if an error is found.

NOTE: The `extloopback` command does not support testing all installed adapters using `n=all`.

To run the external loopback test, type the following command:

```
esxcli linlpcfg loopbacktest external start n=<adapter_number> r=<repeat_count> o=<option_on_error>
```

where:

- `r` = 1 to 4096
- `o` = 1 to 3
  - `1` stops the test on the first error
  - `2` ignores three errors and either stops the test on the fourth error or continues testing the next adapter on the fourth error.
  - `3` ignores errors and continues the test.

The following example runs the external loopback test 50 times on adapter number 1 and stops the test if an error occurs:

```
esxcli linlpcfg loopbacktest external start n=1 r=50 o=1
```

5.2.7.2 Running the Internal Loopback Test (intloopback)

ATTENTION: Do not perform an internal loopback test on an FC adapter that is being used to boot from SAN. The `intloopback` command might cause a loss of connectivity to the SAN and possible loss of data. To perform an `intloopback` command on an FC adapter, make sure the adapter is not currently being used to boot from SAN. There are two ways to perform this action:

- Move the adapter on which you want to perform the internal loopback test onto a non-boot-from-SAN host, and perform the test from there.
- If the host with the test target adapter is also hosting other boot-from-SAN adapters, carry out a boot from SAN using one of the other boot-from-SAN adapters. The `intloopback` command can now be run on the intended target adapter because it is not being actively used for boot from SAN.

This command runs the internal loopback test. You can run the test on a specific adapter in the system. Specify the number of times you want the test to repeat, and indicate the action to be taken if an error occurs.

NOTE: The `intloopback` command does not support testing all installed adapters using `n=all`.

Do not use the internal loopback test on Emulex blade adapters.

To run the internal loopback test on an FC adapter, type the following command:

```
esxcli linlpcfg loopbacktest internal start <n=adapter_number> r=<repeat_count> o=<option_on_error>
```

where:

- `r` = 1 to 4096
- `o` = 1 to 3
  - `1` stops the test on the first error.
  - `2` ignores three errors and either stops the test on the fourth error or continues testing the next adapter on the fourth error.
  - `3` ignores errors and continues the test.

The following example runs the internal loopback test 100 times on FC adapter number 1 and stops the test if an error occurs:

```
esxcli linlpcfg loopbacktest internal start n=1 r=100 o=1
```

5.2.7.3 Running the PCI Loopback Test (pciloopback)

ATTENTION: Do not perform a PCI loopback test on an FC adapter that is being used to boot from SAN. The `pciloopback` command might cause a loss of connectivity to the SAN and possible loss of data. To perform a `pciloopback` command on an FC adapter, make sure the adapter is not currently being used to boot from SAN. There are two ways to perform this action:

- Move the adapter on which you want to perform the PCI loopback test onto a non-boot-from-SAN host, and perform the test from there.
- If the host with the test target adapter is also hosting other boot-from-SAN adapters, carry out a boot from SAN using one of the other boot-from-SAN adapters. The `pciloopback` command can now be run on the intended target adapter because it is not being actively used for boot from SAN.

This command runs the PCI loopback test. You can run the test on a specific adapter or on all adapters in the system. Specify the number of times you want the test to repeat, and indicate the action to be taken if an error occurs.

To run the PCI loopback test, type the following command:

```
esxcli linlpcfg loopbacktest pci start n=<all|adapter_number> r=<repeat_count> o=<option_on_error>
```

where:

- `r` = 1 to 4096
- `o` = 1 to 3
  - `1` stops the test on the first error.
  - `2` ignores three errors and either stops the test on the fourth error or continues testing the next adapter on the fourth error.
  - `3` ignores errors and continues the test.

The following example runs the PCI loopback test 100 times on FC adapter number 2 in the system and stops the testing if any errors occur.

```
esxcli linlpcfg loopbacktest pci start n=2 r=100 o=1
```

5.2.7.4 Viewing Transceiver Data (getxcvrdata)

This command displays transceiver data for an adapter port.

To view transceiver data, type the following command:

```
esxcli linlpcfg xcvrdata get n=<adapter_number>
```

5.2.8 Using Script Files

Custom script files are not supported in the ESXi 8.0 operating system. However, the following script commands are supported:

- `scriptwwnn`
- `scriptwwpn`
- `scriptvwwnn`
- `scriptvwwpn`

5.2.8.1 Updating the Nonvolatile WWNN (scriptwwnn)

ATTENTION: Do not run a `scriptwwnn` command on an FC adapter that is being used to boot from SAN. The `scriptwwnn` command might cause a loss of connectivity to the SAN and possible loss of data. To update the nonvolatile WWNN on an FC adapter, make sure the adapter is not currently being used to boot from SAN. There are two ways to perform this action:

- Move the adapter on which you want to use the `scriptwwnn` command to a non-boot-from-SAN host, and run the script from there.
- If the host with the target adapter is also hosting other boot-from-SAN adapters, carry out a boot from SAN using one of the other boot-from-SAN adapters. You can now run the script to update the nonvolatile WWNN on the intended target adapter because it is not being actively used for boot from SAN.

This command reads the WWNN words 0 and 1 from the command line to update the nonvolatile WWNN. When the adapter is discovered, the new WWNN value is used. The adapter retains the original WWNN in another region of the firmware.

ATTENTION: Use the `scriptwwnn` command with caution. If you use the same WWNN on more than one adapter in a fabric, unpredictable results might occur.

To change WWNN words 0 and 1 from the command line, type the following command:

```
esxcli linlpcfg script WWNN n=<adapter_number> w0=<wwnn_word_0> w1=<wwnn_word_1>
```

The following example updates nonvolatile WWNN word 0 and word 1 for adapter number 1:

```
esxcli linlpcfg script WWNN n=1 w0=10000345 w1=B620A1B2
```

NOTE: Word 0 of WWNN and WWPN names must follow one of the following formats:

```
1 0 0 0 0 x x x
2 x x x x x x x
3 x x x x x x x
5 x x x x x x x
```

If the `scriptvwwnn` command has been used previously, the adapter continues to use that WWNN until you change the WWNN with the `restorenvwwn` command.

5.2.8.2 Updating the Nonvolatile WWPN (scriptwwpn)

ATTENTION: Do not run a `scriptwwpn` command on an FC adapter that is being used to boot from SAN. The `scriptwwpn` command might cause a loss of connectivity to the SAN and possible loss of data. To update the nonvolatile WWPN on an FC adapter, make sure the adapter is not currently being used to boot from SAN. There are two ways to perform this action:

- Move the adapter on which you want to use the `scriptwwpn` command to a non-boot-from-SAN host, and run the script from there.
- If the host with the target adapter is also hosting other boot-from-SAN adapters, carry out a boot from SAN using one of the other boot-from-SAN adapters. You can now run the script to update the nonvolatile WWPN on the intended target adapter because it is not being actively used for boot from SAN.

This command reads WWPN words 0 and 1 from the command line to update the nonvolatile WWPN. When it is discovered, the adapter uses the new WWNN value. It does not use the original IEEE address assigned by manufacturing (located in configuration region 16 or 32).

ATTENTION: Use the `scriptwwpn` command with caution. If you use the same WWPN on more than one adapter in a fabric, unpredictable results might occur.

NOTE: If the `scriptvwwnn` command has been used previously, the adapter continues to use that WWPN until you change the WWPN with the `restorenvwwn` command.

To change WWPN words 0 and 1 from the command line, type the following command:

```
esxcli linlpcfg script WWPN n=<adapter_number> w0=<wwpn_word_0> w1=<wwpn_word_1>
```

The following example updates the nonvolatile WWPN word 0 word 1 for adapter number 1:

```
esxcli linlpcfg script WWPN n=1 w0=20A2D6B8 w1=C920A1B2
```

5.2.8.3 Updating the Volatile WWNN (scriptvwwnn)

ATTENTION: Do not run a `scriptvwwnn` command on an FC adapter that is being used to boot from SAN. The `scriptvwwnn` command might cause a loss of connectivity to the SAN and possible loss of data. To update the volatile WWNN on an FC adapter, make sure the adapter is not currently being used to boot from SAN. There are two ways to perform this action:

- Move the adapter on which you want to use the `scriptvwwnn` command to a non-boot-from-SAN host, and run the script from there.
- If the host with the target adapter is also hosting other boot-from-SAN adapters, carry out a boot from SAN using one of the other boot-from-SAN adapters. You can now run the script to update the volatile WWNN on the intended target adapter because it is not being actively used for boot from SAN.

This command reads the WWNN words 0 and 1 from the command line to update the volatile WWNN. The next time the adapter is discovered, it uses this new WWNN. It does not use the original IEEE address assigned by manufacturing (located in configuration region 16 or 32), nor does it use the value entered by the `writewwn` or `scriptwwnn` commands.

If the adapter does not have firmware that supports the volatile WWN, the following error message appears:

```
Write Volatile Parms Error. Reported Error 48
```

If this error message appears, install firmware that supports the volatile WWN.

NOTE: Word 0 of WWNN and WWPN names must follow one of the following formats:

```
1 0 0 0 0 x x x
2 x x x x x x x
3 x x x x x x x
5 x x x x x x x
```

After you issue this command, the volatile WWNN is used by the adapter until the `restorenvwwn` command is issued or the system is restarted.

ATTENTION: Use the `scriptvwwnn` command with caution. If you use the same volatile WWNN on more than one adapter in a fabric, unpredictable results might occur.

To change volatile WWNN words 0 and 1 from the command line, type the following command:

```
esxcli linlpcfg script VWWNN n=<adapter_number> w0=<wwnn_word_0> w1=<wwnn_word_1>
```

The following example updates the volatile WWNN word 0 and word 1 for adapter number 1:

```
esxcli linlpcfg script VWWNN n=1 w0=20A2D6B8 w1=C920A1B2
```

5.2.8.4 Updating the Volatile WWPN (scriptvwwpn)

ATTENTION: Do not run a `scriptvwwpn` command on an FC adapter that is being used to boot from SAN. The `scriptvwwpn` command might cause a loss of connectivity to the SAN and possible loss of data. To update the volatile WWPN on an FC adapter, make sure the adapter is not currently being used to boot from SAN. There are two ways to perform this action:

- Move the adapter on which you want to use the `scriptvwwpn` command to a non-boot-from-SAN host, and run the script from there.
- If the host with the target adapter is also hosting other boot-from-SAN adapters, carry out a boot from SAN using one of the other boot-from-SAN adapters. You can now run the script to update the volatile WWPN on the intended target adapter because it is not being actively used for boot from SAN.

This command reads the WWPN words 0 and 1 from the command line to update the volatile WWPN. The next time the adapter is discovered, it uses this new WWPN. It does not use the original IEEE address assigned by manufacturing (located in configuration region 16 or 32), nor does it use the value entered by the `writewwn` or `scriptwwpn` commands.

If the adapter does not have firmware that supports the volatile WWPN, the following error message appears:

```
Write Volatile Parms Error. Reported Error 48
```

If this error message appears, install firmware that supports the volatile WWPN.

NOTE: Word 0 of WWNN and WWPN names must follow one of the following formats:

```
1 0 0 0 0 x x x
2 x x x x x x x
3 x x x x x x x
5 x x x x x x x
```

After you issue this command, the volatile WWNN is used by the adapter until the `restorenvwwn` command is issued or the system is restarted.

ATTENTION: Use the `scriptvwwpn` command with caution. If you use the same VWWNN on more than one adapter in a fabric, unpredictable results might occur.

To change VWWPN words 0 and 1 from the command line, type the following command:

```
esxcli linlpcfg script VWWPN n=<adapter_number> w0=<wwpn_word_0> w1=<wwpn_word_1>
```

The following example updates the volatile WWPN word 0 and word 1 for adapter number 1:

```
esxcli linlpcfg script VWWPN n=1 w0=20A2D6B8 w1=C920A1B2
```

Chapter 6: Status Messages and Error Codes

6.1 Elxflash Utility Status Messages

The Elxflash utility takes a less stringent approach to reporting status messages when used with autodiscovery. An error message is returned only when the Elxflash utility reports a failure during a download or if the `firmware` subdirectory cannot be located.

Table 5 lists the supported Elxflash utility status messages.

Table 5: Elxflash Status Messages

Message Number	Message Title	Message Details
0	`GOOD_ALL_UPGRADES_OK`	Returned if any of the discovered adapters had a successful download performed.
1	`GOOD_NO_UPDATES_NEEDED`	Returned if any of the discovered adapters did not need an update, a downgrade, or a rewrite operation. This message is never returned during a force firmware download operation.
2	`ERROR_ALL_UPGRADES_FAILED`	Returned if all attempted downloads failed.
3	`ERROR_SOME_UPGRADES_FAILED`	Returned if some of the attempted downloads failed.
4	`ERROR_EMULEX_APPS_COMMAND`	Returned if Elxflash cannot run an external executable, such as *`oslpcfg`* (`elxflashOffline`).
5	`ERROR_NO_SUPPORTED_HBA_FOUND`	Returned if no supported adapters are found.
6	`ERROR_DIRECTORY_NOT_FOUND`	Returned if the `firmware` subdirectory is missing. Depending on the operation, autodiscovery expects the `firmware` subdirectory to exist as a subdirectory in the root of the `package` directory.
7	`GOOD_NO_UPGRADES_AVAILABLE`	Returned if none of the discovered adapters had a matching image in the `firmware` subdirectory. In this case only, a matching image is an image that the adapter accepts regardless of the download operation being performed.
8	`ERROR_NOT_ADMIN_USER`	Returned when you attempt to run the executable file and do not have administrator (Windows) or root (Linux) privileges.
9	`ERROR_UNSUPPORTED_OS`	Used by the Exlflash Stand-alone execution scripts. Returned when the scripts are run on an unsupported operating system.
10	`ERROR_UNSUPPORTED_ARCH`	Used by the Exlflash Stand-alone execution scripts. Returned when the scripts are run on an unsupported architecture.
19	`ERROR_IMAGE_VERSION`	An error occurred when decoding the image version. The image file has an invalid extension, or Elxflash was unable to decode the image file's version. **NOTE:** This is message returned only when using the `/iv=`*`<image_version>`* switch.
20	`ERROR_MISSING_DEP`	Returned when a missing dependency is detected.
31	`DOWNLOAD_FAILED_INCOMPATIBLE_IMAGE`	A component of the firmware image is incompatible with the HBA. Install a later version of the firmware from www.broadcom.com.
32	`DOWNLOAD_FAILED_INCOMP_FLASH_PART`	A new flash device on the HBA is not supported by the version of firmware you are installing. Install a later version of firmware from www.broadcom.com.
33	`DOWNLOAD_FAILED_INCORRECT_ASIC_VERSION`	The firmware you are installing does not support the HBA. Install a firmware image that supports the HBA from www.broadcom.com.

Table 5: Elxflash Status Messages (Continued)

Message Number	Message Title	Message Details
34	`DOWNLOAD_FAILED_INCOMP_OBJ_GENERAL_ERROR`	The firmware image has an unknown error. Contact Broadcom Technical Support.
35	`DOWNLOAD_FAILED INCOMP_OBJECT`	The firmware image contains an incompatible object. Install a later version of the firmware from www.broadcom.com.

6.2 LpCfg Error Codes

Table 6 lists the supported LpCfg error codes.

Table 6: LpCfg Error Codes

Error Code	Description
0	No error.
1	Invalid adapter number.
2	Mailbox command error.
3	No valid boot (BIOS) code found.
4	Open file error.
5	Invalid configuration region.
6	Invalid adapter name.
7	Download error.
9	Link not up for external loopback test.
10	Link not up for internal loopback test.
13	PCI loopback test fails.
14	Adapter reset error.
15	Read configuration region error.
16	No VPD information available.
17	No command in command line.
18	Open log file error.
19	Read wakeup parameters error.
20	Update wakeup parameters error.
21	Incorrect test parameters.
22	Stopped by user.
23	Internal loopback test fails.
24	External loopback test fails.
25	Error exists after four retries.
26	Invalid command.
27	Incorrect syntax.
28	Command supported only in script files.
29	Read_rev error.
30	Dump configuration region error.
31	Read file error.
32	Short file error.
33	Read NVPARMS error.

Table 6: LpCfg Error Codes (Continued)

Error Code	Description
34	Write NVPARMS error.
35	Command does not support all adapters.
36	Invalid LUN.
37	No boot (BIOS) code enabled.
38	Update configuration region error.
39	No boot (BIOS) found.
40	Dump memory error.
41	Update EROM error.
42	Delete load entry error.
43	Write WWN error.
44	Not supported in script files.
45	No Emulex adapter found.
46	Invalid alternate boot device index.
47	Cannot restart adapter.
48	Write volatile parameters error.
50	Incorrect symbols.
51	Invalid length.
52	Invalid topology.
53	No event log.
54	Read event log.
55	Invalid input value.
56	No libdfc library.
57	Non-numeric input.
58	No valid WWN.
59	Region cleanup.
60	Region initialization.
62	Unable to allocate memory.
63	DFC_InitDiagEnv error.
64	DFC_ReadPciCfg error.
65	No driver installed.
66	No valid driver.
67	Not valid adapter type.
68	Not valid image.
69	Long file error.
70	Incompatible image.
71	Not supported.
73	Command not supported by the adapter's hardware or firmware
78	Administrator rights error.
79	SLI-4 management error.
80	Reboot required.
83	Application is missing one or more dependencies.
84	Unable to retrieve adapter information.

Table 6: LpCfg Error Codes (Continued)

Error Code	Description
87	The system must be rebooted to activate the new firmware image, and the image must be downloaded again.
89	Download failed due to a missing digital signature. Contact Broadcom Technical Support for additional help. Download failed on adapter `<adapter_number>` Stat 89.
90	Download failed due to an invalid firmware digital signature. Contact Broadcom Technical Support for additional help. Download failed on adapter `<adapter_number>` Stat 90.
91	Error while executing factory reset.
92	This firmware version is not supported on this board model.
93	The `linlpcfg` executable has missing dependencies.
94	Port synchronization error. Reset all ports on this adapter, and then retry the download operation.
95	Adapter PCI reset failed.
96	The new firmware is activated. Some features require an optional reboot. Refer to the adapter's firmware and boot code release notes for details.
97	The OS is not supported.
103	The flash device on the HBA is not supported by the version of firmware you are installing. Install a later version of firmware from www.broadcom.com.
104	The firmware you are installing does not support the HBA. Install a firmware image that supports the HBA from www.broadcom.com.
105	The firmware image has an unknown error. Contact Broadcom Technical Support.
106	A component of the firmware image is incompatible with the HBA. Install a later version of the firmware from www.broadcom.com.
200	General error.

Chapter 7: Troubleshooting

When using Elxflash, some commands take precedence over others. For instance, the force firmware (`/ff`) switches take precedence over the `/downgrade`, `/update`, and `/rewrite` switches. If `/ff` is used, the `/downgrade`, `/update`, and `/rewrite` switches are ignored.

The following examples illustrate this behavior:

- `./elxflash /ff /update`
  - `/update` is ignored.
  - Performs a force firmware operation on all installed and supported adapters.
- `./elxflash /ff /downgrade`
  - `/downgrade` is ignored.
  - Performs a force firmware operation on all installed and supported adapters.
- `./elxflash /ff /rewrite`
  - `/rewrite` is ignored.
  - Performs a force firmware operation on all installed and supported adapters.

7.1 Unsupported Drivers

The Elxflash utility reports an error similar to the following if an unsupported driver is installed on the system:

```
elxflash: no supported Emulex HBA's found - Return Code=1
```

If an error similar to this example occurs, verify that the correct version of the driver is installed. For supported adapters and supported versions of operating systems and platforms, go to www.broadcom.com.

NOTE: The Linux Elxflash Offline and Elxflash Stand-alone kits require out-of-box FC drivers. If the out-of-box FC drivers are not installed, LpCfg and Elxflash report incorrect data for FC functions.

The Windows Elxflash Offline and Elxflash Stand-alone kits require out-of-box FC drivers. If the out-of-box FC drivers are not installed, the following events occur:

- FC functions are not discovered.
- LpCfg and Elxflash report incorrect data for FC functions.
- Firmware downloads to FC functions might not be available.

7.2 Erroneous Output

If the Elxflash utility reports erroneous output, run the application with the `/xv` switch. The `/xv` switch creates a log file called `dbg.log`. Forward this log to Broadcom Technical Support.

7.3 Firmware Download Failures

On Windows operating systems, a firmware download failure might not generate the correct error message. In such a case, contact an authorized Broadcom Technical Support representative.

BROADCOM®