Emulex® Elxflash and LpCfg Utilities

User Guide
Release 12.8

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

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 and boot code download solutions for the following Emulex FC adapters in production systems:

- LPe12000-series adapters
- LPe16000-series adapters
- LPe31000-series adapters
- LPe32000-series adapters
- LPe35000-series adapters

You can also download firmware and boot code 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), `/fb` (force boot), `/update`, `/downgrade`, and `/rewrite`.

When using autodiscovery, you must populate the firmware and boot subdirectories with image files for specific adapter models. Autodiscovery uses Emulex defined firmware and boot code file names in their respective directories.

If you do not want to use autodiscovery, the Elxflash utility can update the firmware or boot code of an adapter using the `fwmatrix.txt` configuration file. When using the `fwmatrix.txt` file, you must update each supported adapter's type, firmware, and boot code entries, and place the corresponding firmware and boot code images into their respective directories. 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:

- `./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, see www.broadcom.com.

1.5 Abbreviations

BFS	boot from SAN
D_ID	destination identifier
FABL	fabric-assigned boot LUN
FCP	Fibre Channel Protocol
ms	milliseconds
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, and Windows operating systems. It also describes the update procedure for each operating system.

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

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 VMware executable files are extracted to the `esxi\` directory, and the Windows executable files are extracted to the `win\` directory.

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

In Windows, for example, three directories are visible in the kit's root directory after you extract the stand-alone utility. The top-level boot directory and firmware directory are the default directories in which Elxflash looks for boot code and firmware images. Ensure that firmware image and boot code are located in these directories.

```
# dir

07/05/2012  07:02 PM    <DIR>          .
07/05/2012  07:02 PM    <DIR>          ..
07/02/2012  03:14 PM    <DIR>          boot
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 VMware 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

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`. For all LPe16000-series, LPe31000-series, LPe32000-series, and LPe35000-series adapters, the firmware image file includes boot code. LPe12000-series adapters use a separate boot code image file, which is expected to be in the subdirectory named `boot`.

Two supported modes for updating firmware and boot code are available for each Elxflash utility switch. The `fwmatrix.txt` file mode allows you to specify which adapter models and firmware file (or firmware and boot files) to update. Autodiscovery mode automatically updates the discovered adapters with the contents of the `firmware` directory (and the `boot` directory, if applicable).

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

For example:

```
# ./elxflash /q
HBA=LPe32000, Port Type=FC, WWN=10:00:00:90:FA:94:2E:CA, PCI ID=E300, VID=10DF, SSID=E321, SVID=10DF,
Firmware=11.2.50.48, Boot Code=11.2.50.35, Boot Enabled=1
HBA=LPe12002, Port Type=FC, WWN=10:00:00:00:C9:A1:80:00, PCI ID=F100, VID=10DF, SSID=F100, SVID=10DF,
Firmware=UD202A1, Boot Code=UU700A2, Boot Enabled=1
HBA=LPe12002, Port Type=FC, WWN=10:00:00:00:C9:A1:80:01, PCI ID=F100, VID=10DF, SSID=F100, SVID=10DF,
Firmware=UD202A1, Boot Code=UU700A2, Boot Enabled=1
```

If the `/q` switch is used with an operational switch, the results include an additional field called `Supported Firmware` or `Supported Boot Code`.

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

- FC adapter download summary:

  ```
  <date><time>
  HBA=<model>, Port Type=<port_type>, WWN=<wwn>,
  Update=<Boot Code|Firmware>, 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 and boot code directories with the appropriate firmware and boot code images. The `fwmatrix.txt` file includes adapter entries followed by a firmware image file entry. An additional boot image entry is provided for LPe12000-series adapters.

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 adapter model entry and then attempts to download the specified firmware image file and, for LPe12000-series adapters, the specified boot code image file.

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` and `boot` subdirectories, 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` and `boot` directories.

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

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

Example usage:

`./elxflash /auto /update /q` – Updates the firmware and boot code using the `firmware` and `boot` directories.

- The `fwmatrix.txt` file is ignored. The desired versions of firmware must be in the `firmware` directory.
- Using the `firmware` subdirectory, the Elxflash utility automatically discovers the best matching firmware for each installed and supported adapter.
- If multiple versions of firmware or boot code 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 or boot code of each adapter if the currently installed versions are more recent than the downgrade versions. This switch cannot be used with the `/update` or `/rewrite` commands.

Example usage:

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

- The `fwmatrix.txt` file is ignored. The desired downgrade versions of firmware or boot code must be in their respective directories.
- 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 or boot code are found for an adapter, the next-previous downgrade versions are downloaded to the adapter.

- When performing the boot code downgrade operation, the Elxflash utility first tries to match by adapter family and boot type. If a match is not found, the Elxflash utility then tries to match by boot type. If the utility matches by boot type and multiple versions of boot code are detected, the utility chooses the downgrade file in the following order:
  - Universal (U)
  - Pair (P)
  - Open (O)
  - EFI (E)
  - x86 (B)

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

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

3.5 Force Firmware and Boot Code (/f)

Usage: `/f`

The `/f` switch forces a firmware and boot code 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 Boot Code (/fb), and Section 3.7, Force Firmware (/ff).

Example usage:

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

- The `fwmatrix.txt` file is ignored. The desired versions of firmware and boot code must be in their respective directories.
- If multiple versions of firmware or boot code are found for an adapter, the Elxflash utility uses the most recent version when performing the firmware and boot code downloads.

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

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

3.6 Force Boot Code (/fb)

Usage: `/fb`

The `/fb` switch forces a boot code download to an adapter regardless of the boot code version installed on the adapter. The switch is performed regardless of any additional operational switches given on the command line.

Example usage:

`./elxflash /fb /auto /q` – Forces a boot code download using autodiscovery.

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

`./elxflash /fb /q` – Forces a boot code download using the `fwmatrix.txt` file.

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

NOTE: The `/fb` switch applies only to LPe12000-series adapters.

3.7 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.

NOTE: Because boot code is included in the firmware image for LPe16000-series, LPe31000-series, LPe32000-series, and LPe35000-series adapters, this command has the same result as the `/f` command, which forces a firmware and boot code download.

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` directory.
- 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.

3.8 Firmware Matrix Directory Change (/fmd)

Usage: `/fmd=<`*`directory`*`>`

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

3.9 Help (/h or /?)

Usage: `/h or /?`

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

3.10 Image Directory Change (/id)

Usage: `/id=<`*`image_directory`*`>`

The `/id` switch specifies the location of the `firmware` and `boot` directories.

Example usage:

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

- Elxflash looks for the `firmware` and `boot` directories in the `/tmp` directory.
- The `fwmatrix.txt` file is ignored. The desired versions of firmware and boot code must be in their respective directories.
- In this example, firmware must be placed in the `/tmp/firmware` directory, and boot code must be placed in the `/tmp/boot` directory.
- If multiple versions of firmware or boot code are found for an adapter, Elxflash uses the most recent versions when performing the firmware and boot code downloads.

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

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

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

3.11 Image Version Display (/iv)

Usage: `/iv=<image_`*`path`*`>/<`*`image_file`*`>`

The `/iv` switch 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/A11460.grp
1.1.46.0
```

3.12 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.

3.13 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:

- `/ff`
- `/fb`
- `/downgrade`
- `/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 or boot code using autodiscovery.

- The `fwmatrix.txt` file is ignored. The desired update versions of firmware or boot code must be in their respective directories.
- 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 or boot code are found for an adapter, the Elxflash utility provides a download preview using the most recent versions.

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

- For each installed and supported adapter, the current firmware and boot code versions are compared with 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 or boot code for each adapter that can be updated.

3.14 Process FC HBAs Only (/fc)

Usage: `/fc`

Use the `/fc` switch on FC HBAs only.

Example usage:

`./elxflash /q /fc` – Only FC HBAs are displayed for a query.

`./elxflash /auto /up /fc` – This command applies auto-update to FC HBAs only.

When a firmware or boot code update is performed and the `/fc` switch is used, only FC adapters are updated.

3.15 Query (/q)

Usage: `/q`

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

Example usage:

```
C:\elxflashStandalone-windows-10.0.567.22-1\win>elxflash.bat /q
HBA=<model>, Port Type=<port_type>, WWN =<wwn>, Firmware=<version>, Boot Code=<version>
```

NOTE: On FC adapters, the query switch displays boot code version information only if the adapter has boot code installed.

3.16 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.

3.17 Rewrite (/rewrite or /e)

Usage: `/rewrite` or `/e`

The `/rewrite` switch updates the firmware or boot code 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 or boot code using autodiscovery.

- The `fwmatrix.txt` file is ignored. The desired rewrite versions of firmware or boot code must be in their respective directories.
- 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 or boot code are found for an adapter, the most recent versions are downloaded to the adapter.
- When performing the boot code rewrite operation, Elxflash first tries to match by adapter family and boot type. If a match is not found, Elxflash then tries to match by boot type. If the utility matches by boot type, and multiple versions of boot code are detected, rewrite always chooses the boot code in the following order:
  - Universal (U)
  - Pair (P)
  - Open (O)
  - EFI (E)
  - x86 (B)

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

- For each installed and supported adapter, the current firmware or boot code versions are compared with the versions 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 or boot code are downloaded to the adapter.

3.18 Silent (/s)

Usage: `/s`

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

Example usage:

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

- The `fwmatrix.txt` file is ignored. The desired versions of firmware and boot code must be in their respective directories.
- If multiple versions of firmware or boot code are found for an adapter, Elxflash uses the most recent versions when performing the firmware and boot code downloads.
- No output is printed to stdout.

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

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

3.19 Update (/update)

Usage: `/update`

The `/update` switch updates the firmware or boot code 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 or boot code using autodiscovery.

- The `fwmatrix.txt` file is ignored. The desired update versions of firmware or boot code must be in their respective directories.
- 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 or boot code are found for an adapter, the most recent versions are downloaded to the adapter.
- When performing the boot code update operation, the Elxflash utility uses the most recent boot code version found. Because multiple compatible versions of boot code can exist, the most recent version is selected in the following order:
  - Universal (U)
  - Pair (P)
  - Open (O)
  - EFI (E)
  - x86 (B)

NOTE: The most recent boot code version is downloaded regardless of the installed boot code. This download allows upgrading from one type of boot code to another type.

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

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

3.20 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

3.21 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` or `/g`
- `/f` – Force firmware and boot
- `/fb` – Force boot code
- `/ff` – Force firmware
- `/query` or `/q`
- `/rewrite` or `/e`
- `/update`
- `/xml` – XML output

NOTE: The `/vpd` command does not display any information if 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.

3.22 XML Output (/xml)

Usage: `/xml`

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

Chapter 4: LpCfg Utility Command Line Interface

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

The LpCfg utility includes:

- 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 do the following:

- View information about an Emulex adapter
- Reset the adapter
- Return the adapter to its factory default settings
- Update firmware and boot code files
- Select a boot device
- Read and update WWNs
- Enable and configure boot code for FCP and NVMe over FC
- Update configuration regions
- Run diagnostic tests
- Read and process script files
- Read, write, and reset VLAN IDs and VLAN priorities

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 4.3.9, 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:

   - `boot\`
   - `firmware\`
   - `lx\`
3. For Elxflash, copy the firmware images to the `firmware` directory. Copy the boot images to the `boot` directory.

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:

   - `i386\`
   - `ppc64\`
   - `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: The Emulex OneCommand® Manager application has been renamed as the Emulex HBA Manager application.

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 and boot code 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 `Elxflash Standalone-windows-<`*`version`*`>`.

   The following directories must be present:

```
– boot\
– firmware\
– win\
```

3. For Elxflash, copy the firmware images to the `firmware` directory. Copy the boot images to the `boot` directory.

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. Run Elxflash, and revert any changes before exiting.

The `winlpcfg.bat` script configures the environment. Run winlpcfg, and revert 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 and boot code 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 1 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 1: Supported Commands for LpCfg

	FC Adapters	
Commands	**x64**	**IBM PowerPC® HBAs**
Operating Systems	RHEL 7.7+ RHEL 8.0+	RHEL 7.7+ RHEL 8.0+
	SLES 12 SP4+ SLES 15 SP1+	SLES 12 SP4+ SLES 15 SP1+
	ESXi 6.5 ESXi 6.7 ESXi 7.0	N/A
	WinPE 10.1 (Dell only)	N/A
`config`	✓	L
`directDownload` Supported on LPe12000-series adapters only.	L	L
`disableboot` (deprecated command)	N/A	N/A
`disablebootdevice` (deprecated command)	N/A	N/A
`download` The `a=<adapter name>` parameter is supported on LPe12000-series adapters only.	✓	L
`enableboot` (deprecated command)	N/A	N/A
`enablebootdevice` (deprecated command)	N/A	N/A
`extloopback`	✓	L
`factorydefaults`	✓	L
`getbootparams` (new command)	✓	L
`getfwparams` (new command)	✓	L
`getxcvrdata` (new command) Supports LPe31000-series, LPe32000-series, and LPe35000-series adapters only	✓	L
`h (Help)`	✓	L
`hbaattr`	✓	L
`intloopback`	✓	L
`listboot`	✓	L

Table 1: Supported Commands for LpCfg (Continued)

Commands	FC Adapters	
	x64	IBM PowerPC® HBAs
`listhba`	✓	L
`listrev`	✓	L
`listwwn`	✓	L
`logfile`	✓	L
`pciloopback`	✓	L
`posttest` Supported on LPe12000-series adapters only.	✓	L
`readaltboot` (deprecated command)	N/A	N/A
`readbootdevice` (deprecated command)	N/A	N/A
`readconfig`	✓	L
`reset`	✓	L
`restoredefwwn`	✓	L
`restorenwwn`	✓	L
`restorewwn`	✓	L
`savewwn`	✓	L
`screendisplay`	✓	L
`scriptvwwnn`	✓	L
`scriptvwwpn`	✓	L
`scriptwwnn`	✓	L
`scriptwwpn`	✓	L
`setaltboot` (deprecated command)	N/A	N/A
`setbootdevice` (deprecated command)	N/A	N/A
`setbootparam` (new command)	✓	L
`setfwparam` (new command)	✓	L
`version`	✓	L
`vpd`	✓	L
`writewwn`	✓	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. You can also run a POST.

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>
  ```

  or

  ```
  download a=<adapter_type> i=<path/image_filename>
  ```

- For Windows:

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

  or

  ```
  download a=<adapter_type> 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 on LPe35000-series, LPe32000-series, LPe31000-series, or LPe16000-series adapters, the adapter is offline. Perform an immediate reboot to complete the reset and return the adapter to full functionality. An LPe12000-series adapter is online after the reset is complete and does not require a reboot.

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
```

4.3.1.4 Running a Power-On Self-Test (posttest)

This command runs a POST on the selected adapter.

ATTENTION: Do not run a POST on an FC adapter that is being used to boot from SAN. The `posttest` command might cause a loss of connectivity to the SAN and possible loss of data. To run a POST 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 run the POST to a non-boot-from-SAN host, and run the POST from there.
- If the host with the `posttest` 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 POST can now be run on the intended target adapter because it is not being actively used for boot from SAN.

To run the adapter POST, type the following command:

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

The following example runs a POST on adapter number 1.

```
oslpcfg posttest 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 Emulex Conventional Names Instead of the VPD Model Name (/c)

Adding `/c` to any command that uses the `a=` parameter to return adapter data causes the command to return an Emulex conventional model for the adapter rather than the VPD model name of the adapter. The `/c` switch can be used for any command that uses the model name. For instance, the following commands use the `a=` parameter and show this behavior.

- `config`
- `download`
- `directdownload`
- `listhba`

NOTE: The offline utility does not always display the model name of the adapter being tested.

In the following example, the VPD model name is reported for each adapter:

```
# ./linlpcfg listhba
Command: listhba
HBA 1: 10000090 FA942ECA Functional FW: 11.2.50.48 devID:E300 Bus:8 Dev:0 Func:0 LPe32000-M2-D
HBA 2: 10000000 C9A18000 Functional FW: US2.02A1  devID:F100 Bus:E Dev:0 Func:0 LPe12002-M8
HBA 3: 10000000 C9A18001 Functional FW: US2.02A1  devID:F100 Bus:E Dev:0 Func:1 LPe12002-M8
```

To list adapter information using the conventional name rather than the VPD model name, type the following command:

```
# ./linlpcfg listhba /c
Command: listhba /c
```

```
HBA 1: 10000090 FA942ECA Functional FW: 11.2.50.48 devID:E300 Bus:8 Dev:0 Func:0 LPe32000
HBA 2: 10000000 C9A18000 Functional FW: US2.02A1  devID:F100 Bus:E Dev:0 Func:0 LPe12002
HBA 3: 10000000 C9A18001 Functional FW: US2.02A1  devID:F100 Bus:E Dev:0 Func:1 LPe12002
```

The output from this command includes the Emulex conventional model name (for example, LPe12002), rather than the VPD model name (for example, LPe12002-M8).

4.3.2.2 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
```

4.3.2.3 Viewing the VPD (vpd)

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

To display the VPD, type the following command:

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

4.3.2.4 Viewing Boot Device Information (readbootdevice)

NOTE: This command is deprecated. Use the `getbootparams` command to view boot device information. This section is provided for comparison purposes only.

This command shows the WWN, the LUN (in decimal format), and the topology in use for the currently selected boot device.

To show this information, type the following command:

```
oslpcfg readbootdevice n=<adapter_number>
```

The following example reads WWN and LUN for adapter number 1:

```
oslpcfg readbootdevice n=1
```

4.3.2.5 Viewing Boot Code Versions (listboot)

This command lists all the boot code versions that are loaded in the flash of the adapter. If the selected adapter does not have boot code, it returns error code 39.

To list boot code versions, type 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
```

4.3.2.6 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. For LPe35000-series, LPe32000-series, and LPe31000-series adapters, 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.

4.3.2.7 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
```

NOTE: The `listhba` command with the `/c` option displays the conventional model names instead of the model names contained in the VPD.

4.3.2.8 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 either the `writewwn` command or the `scriptwwpn` and `scriptwwnn` commands. For more information on the `writewwn` command, see Section 4.3.5.1, Writing the WWN and Updating the NVPARMS (writewwn).

If the system does not have any Emulex adapters installed, it returns error code 45.

To show the WWN information, type the following command:

```
oslpcfg listwwn
```

4.3.2.9 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
```

4.3.2.10 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). Valid region numbers are 0 to 32. You must initialize the configuration region first by writing data to it.

NOTE: The `readconfig` command supports reading configuration regions 0, 8, and 32 only on the following adapters:

- LPe16000-series adapters
- LPe31000-series adapters
- LPe32000-series adapters
- LPe35000-series adapters

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
```

4.3.3 Firmware Commands

4.3.3.1 Viewing Firmware Parameters (getfwparams)

This command displays the topology parameter for the following HBAs:

- LPe16000-series
- LPe31000-series
- LPe32000-series
- LPe350000-series

NOTE: To view topology for LPe12000-series HBAs, use the `getbootparams` command.

To view the topology parameter using the `getfwparams` command, type the following command:

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

4.3.3.2 Setting Firmware Parameters (setfwparam)

This command configures the topology parameter for the following HBAs:

- LPe16000-series
- LPe31000-series
- LPe32000-series
- LPe350000-series

NOTE: To configure the topology for LPe12000-series HBAs, use the `setbootparam` command with the `topology` parameter. See Section 4.3.6.1, Configuring Boot Parameters (setbootparam).

To set the topology using the `setfwparam` command, 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 LPe16000-series, LPe31000-series, and LPe32000-series adapters. Point-to-point is the only topology supported on LPe35000-series adapters.

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

4.3.4 Firmware and Boot Code Download Commands

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

4.3.4.1 Downloading a File (download)

This command downloads a firmware or boot code file to a specific adapter.

ATTENTION: Do not download a firmware or boot code file to an FC adapter that is being used to boot from SAN. The download command might cause a loss of connectivity to the SAN and possible loss of data. To download a firmware or boot code file to 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 download the file onto a non-boot-from-SAN host, and perform the download command from there.
- If the host with the download 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 firmware or boot code file can now be downloaded onto the intended target adapter because it is not being actively used for boot from SAN.

The adapter name is the name that appears when you run the `listhba` command. For more information on the `listhba` command, see Section 4.3.2.7, 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 `uu513a10.prg` boot code file to adapter number 6; in this example, the boot code file is for an LPe12000 adapter:

```
oslpcfg download n=6 i=uu513a10.prg
```

NOTE: Boot code on LPe16000-series, LPe31000-series, LPe32000-series, and LPe35000-series adapters is updated by downloading firmware. LpCfg does not support updating boot code separately for these adapters.

To verify the reset or boot status of new firmware, issue the `hbaatrib` command and locate the `Firmware Status` information. See Section 4.3.2.6, Viewing Adapter Attributes (hbaattr), for more information.

For LPe35000-series, LPe32000-series, and LPe31000-series adapters only:

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 SP3 and later
- SLES 15 and later
- RHEL 7.6 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.

4.3.4.2 Accessing the Flash Device Directly (directdownload)

This command accesses the flash device on the adapter directly without using the adapter firmware. This feature is useful in downloading a ROM file image if the firmware has been corrupted. The adapter name is the name that appears when you run the `listhba` command. You can also use `default` for the adapter name if only one single-port adapter or one dual-port adapter is in the system. For more information on the `listhba` command, see Section 4.3.2.7, Viewing All Adapters in the System (listhba).

NOTE: You cannot use `directdownload` in a script file.

This command is supported only on LPe12000-series adapters.

ATTENTION: Do not access the flash device directly on an FC adapter that is being used to boot from SAN. The `directdownload` command might cause a loss of connectivity to the SAN and possible loss of data. To run the `directdownload` 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 run the `directdownload` command to a non-boot-from-SAN host, and run the command from there.
- If the host with the `directdownload` 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 `directdownload` command can now be run on the intended target adapter because it is not being actively used for boot from SAN.

To access the flash device on the adapter directly, type one of the following commands:

In Windows:

```
oslpcfg directdownload a=<adapter_name|default> i=<path\image_filename> s=<0|1>
```

In Linux:

```
oslpcfg directdownload a=<adapter_name|default> i=<path/image_filename> s=<0|1>
```

where `s=1` saves the existing VPD.

The following example accesses the flash device on an LPe12000 HBA in Windows:

```
winlpcfg directdownload a=lpe12000 i=C:\image\ud100a8.rom s=1
```

The following example accesses the flash device if the offline utility cannot detect the adapter type and only one single-port adapter or one dual-port adapter is in the Windows system:

```
winlpcfg directdownload a=default i=C:\image\ud100a8.rom s=1
```

ATTENTION: If `s=0`, the ROM images used with the `directdownload` command might not contain certain VPD information; for example, serial number, adapter model, or manufacturer. Direct download of a ROM image that has not been confirmed to contain the correct VPD image updates the board's firmware, but it also clears the VPD. The board will function. If you use calls for VPD in your applications, the information might be changed or missing.

4.3.5 World Wide Name Commands

The following commands allow you to use the WWN to update the NVPARMS, 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 Writing the WWN and Updating the NVPARMS (writewwn)

This command allows you to enter word 0 and word 1 of the WWPN or WWNN from the keyboard or from a barcode scanner to update a specified adapter's NVPARMS with a new WWPN or WWNN. The new WWPN and WWNN are used the next time the adapter is discovered. The adapter stores the original WWPN and WWNN in another region of the memory so it can identify the adapter as it was manufactured. The WWN can also be read with a barcode scanner.

ATTENTION: Do not run a `writewwn` command on an FC adapter that is being used to boot from SAN. The `writewwn` command might cause a loss of connectivity to the SAN and possible loss of data. To write to the WWN and update 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 `writewwn` command to a non-boot-from-SAN host, and run `writewwn` from there.
- If the host with the `writewwn` 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 write the WWN or update the NVPARMS on the intended target adapter because it is not being actively used for boot from SAN.

The `writewwn` command prompts you for the WWPN and WWNN data words, so it cannot be used in a script file. The `scriptwwnn` and `scriptwwpn` commands use values entered with the command, so they can be used in a script file.

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

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
```

To modify the WWPN and WWNN, type the following command:

```
oslpcfg writewwn n=<adapter_number>
```

The offline utility prompts you to enter new data:

- `WWPN word 0`
- `WWPN word 1`
- `WWNN word 0`
- `WWNN word 1`

The following example writes the WWPN and WWNN for adapter number 1.

```
oslpcfg writewwn n=1
Enter or Scan value for WWPN word 0 now
```

Enter a value.

```
10000000
```

The system echoes what you entered, followed by the next prompt:

```
10000000
Enter or Scan value for WWPN word 1 now
```

4.3.5.2 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
```

4.3.5.3 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
```

4.3.5.4 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 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
```

4.3.5.5 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 `restoredefwwn` 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

NOTE: New commands have replaced deprecated commands in this section.

In earlier versions of the LpCfg CLI, you used the `enableboot` command to enable boot code on an HBA if necessary. Then you issued the `setbootdevice` to enable the HBA to boot from SAN, and you issued the `setaltboot` command to set alternate boot devices. Beginning with this release, you use the `setbootparam` command to perform these steps.

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 4.3.2.5, Viewing Boot Code Versions (listboot), for more information.

   Example:

   ```
   oslpcfg listboot n=1
   ```

   Sample return for an LPe12000-series adapter with x86 BootBIOS enabled:

   ```
   BootBIOS 1 (enabled): UU11.40A2
   Command completed, NO Error
   ```

Sample return for an LPe12000-series adapter with x86 BootBIOS disabled:

```
BootBIOS 1 (disabled): UU11.40A2
Command completed, NO Error
```

Sample return for an LPe16000-series, LPe31000-series, LPe32000-series, or LPe35000-series adapter with x86 BootBIOS enabled:

```
Boot Code (enabled)          = 11.4.120.0
fc_universal_bios_version    = 11.4.120.0
fc_x86_bios_version          = 11.4.113.0
fc_efi_bios_version          = 11.4.120.0
fc_fcode_version             = 11.4.13.0
Command completed, NO Error
```

Sample return for an LPe16000-series, LPe31000-series, LPe32000-series, or LPe35000-series adapter with x86 BootBIOS disabled:

```
Boot Code (disabled)         = 11.4.120.0
fc_universal_bios_version    = 11.4.120.0
fc_x86_bios_version          = 11.4.113.0
fc_efi_bios_version          = 11.4.120.0
fc_fcode_version             = 11.4.13.0
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 to verify the configuration. See Section 4.3.2.4, Viewing Boot Device Information (readbootdevice), for more information.
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)

NOTE: This command replaces the following deprecated commands:

- `enableboot`
- `disableboot`
- `enablebootcode`
- `disablebootcode`
- `setbootdevice`
- `setaltboot`

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

If you are changing boot device parameters for OpenBoot, the `bootdev` keyword is not required.

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 2, Boot Parameters): ■ x86: x86Boot ■ efi: UEFIBoot ■ ob: OpenBoot
param	See Table 2, Boot Parameters.
param_value	See Table 2, Boot Parameters.
bootdev_value	Boot device entry number (0 through 7).

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

Table 2: Boot Parameters

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
	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 2: Boot Parameters (Continued)

Parameter Type	Parameter Name	Description	Supported Boot Types	param_value
Adapter (Continued)	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 or 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
	defaultalpa	The defaultalpa parameter allows you to change the default AL_PA from the standard default of 0x.	x86 and ob LPe12000-series and LPe16000-series HBAs only	See Table 3 for a list of default AL_PAs.

Table 2: Boot Parameters (Continued)

Parameter Type	Parameter Name	Description	Supported Boot Types	param_value
Adapter (Continued)	delaydevicediscovery	The delaydevicediscovery parameter sets a delay to occur after a loop initialization and before a scan is initiated to discover the target. The default value is 0. Change the default if you have an HP MSA1000 or HP MSA1500 RAID array and if both of the following conditions exist: ■ The MSA array is direct-connected or is part of an arbitrated loop (for example, daisy-chained with a JBOD). ■ The boot LUN is not reliably discovered. In this case, a delay might be necessary to allow the array to complete a reset. If both of the conditions exist, typically set this parameter to 20 seconds (0x14). However, the delay needs to be only long enough for the array to be reliably discovered after a reset. Your value might be different. **NOTE:** Do not change the delay device discovery time if your MSA array is connected to a fabric switch. Setting it to any other time guarantees that the maximum delay time is seen on every loop initialization.	efi LPe12000-series HBAs only	0x00 through 0xFF
	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 2: Boot Parameters (Continued)

Parameter Type	Parameter Name	Description	Supported Boot Types	param_value
Adapter (Continued)	`enableadapterboot`	The `enableadapterboot` parameter enables or disables the boot code for the specified adapter. It replaces the deprecated `enableboot` and `disableboot` commands. 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. Changes made to `x86` are also applied to `efi`. Changes made to `efi` are also applied to `x86`.	`x86`, `efi`, and `ob`	`0`=Disable `1`=Enable
	`enablebootfromsan`	The `enablebootfromsan` parameter enables the adapter to boot from SAN. It replaces the deprecated `enablebootdevice` and `disablebootdevice` commands. **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` parameters.	`x86`, `efi`, and `ob`	`0`=Disable `1`=Enable
	`enablefabl`	The `enablefabl` parameter enables Brocade® FABL. Refer to the Brocade switch documentation for more information. **NOTE:** Switch configuration is required when using FABL. Refer to the Brocade switch documentation for configuration details. This parameter is not available on LPe12000-series adapters.	`efi`	`0`=Disable `1`=Enable
	`envvarenable`	The `envvarenable` parameter sets the boot controller order if the system supports the environment variable. The default setting is disabled.	`x86`	`0`=Disable `1`=Enable
	`maxlunspertarget`	The `maxlunspertarget` sets the maximum number of LUNs that are polled during device discovery.	`efi`	`1` through `4096`

Table 2: Boot Parameters (Continued)

Parameter Type	Parameter Name	Description	Supported Boot Types	param_value
Adapter (Continued)	`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`, `efi`, and `ob`	`0`=None `1`=50 ms `2`=100 ms `3`=200 ms
	`sfsflag`	The `sfsflag` parameter enables the inbox Emulex driver for Solaris (also called the SFS driver).	`ob`	`0`=Disable `1`=Enable
	`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
	`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

Table 2: Boot Parameters (Continued)

Parameter Type	Parameter Name	Description	Supported Boot Types	param_value
Adapter (Continued)	topology	The topology parameter allows you to select the topology for the specified LPe12000-series HBA. **NOTE:** Use the setfwparam command to set the topology for all other HBAs. Changes made to x86 are also applied to efi. Changes made to efi are also applied to x86.	x86, efi, and ob	0=Auto (loop first) 1=Auto (point-to-point first) 2=Loop 3=Point-to-point
Boot device	d_id	Specify the boot device by its device ID.	x86 and ob	Use the D_ID of the boot device (in decimal)
	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, efi, and ob	0 through 255
	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, efi, and ob	Use the target WWPN in the following format: nn:nn:nn:nn:nn:nn:nn:nn
	targetid	Specify the boot device by its target ID.	ob	Use the target ID of the boot device (in decimal)
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 parameters.	efi	0=Disable 1=Enable
NVMe boot device	namespaceid	Specify the boot device by its NVMe namespace ID value.	efi	0 through 0xFFFFFFFF
	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
	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
	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.

The following table provides the default AL_PAs that you can set using the defaultalpa parameter.

Table 3: Valid AL_PA Values

0x00	0x01	0x02	0x04	0x08	0x0F	0x10	0x17
0x18	0x1B	0x1D	0x1E	0x1F	0x23	0x25	0x26
0x27	0x29	0x2A	0x2B	0x2C	0x2D	0x2E	0x31
0x32	0x33	0x34	0x35	0x36	0x39	0x3A	0x3C
0x43	0x45	0x46	0x47	0x49	0x4A	0x4B	0x4C
0x4D	0x4E	0x51	0x52	0x53	0x54	0x55	0x56
0x59	0x5A	0x5C	0x63	0x65	0x66	0x67	0x69
0x6A	0x6B	0x6C	0x6D	0x6E	0x71	0x72	0x73
0x74	0x75	0x76	0x79	0x7A	0x7C	0x80	0x81
0x82	0x84	0x88	0x8F	0x90	0x97	0x98	0x9B
0x9D	0x9E	0x9F	0xA3	0xA5	0xA6	0xA7	0xA9
0xAA	0xAB	0xAC	0xAD	0xAE	0xB1	0xB2	0xB3
0xB4	0xB5	0xB6	0xB9	0xBA	0xBC	0xC3	0xC5
0xC6	0xC7	0xC9	0xCA	0xCB	0xCC	0xCD	0xCE
0xD1	0xD2	0xD3	0xD4	0xD5	0xD6	0xD9	0xDA
0xDC	0xE0	0xE1	0xE2	0xE4	0xE8	0xEF	—

4.3.6.2 Enabling or Disabling Boot Code (enableboot or disableboot)

NOTE: These commands are deprecated. Use the `setbootparam` command with the `enableadapterboot` parameter to enable or disable the boot code. This section is provided for comparison purposes only.

This command enables or disables the boot code for the specified adapter.

ATTENTION: Do not enable or disable the boot code on an FC adapter that is being used to boot from SAN. The `enableboot` or `disableboot` command might cause a loss of connectivity to the SAN and possible loss of data. To run either of these commands 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 enable or disable the boot code to a non-boot-from-SAN host, and run the command 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. The `enableboot` or `disableboot` command can now be run on the intended target adapter because it is not being actively used for boot from SAN.

To enable boot code, type the following command:

```
oslpcfg enableboot n=<adapter_number> i=1
```

NOTE: Always set `i` to `1`.

The following example enables boot code on adapter number 6:

```
oslpcfg enableboot n=6 i=1
```

To disable boot code, type the following command:

```
oslpcfg disableboot n=<adapter_number>
```

The following example disables boot code on adapter number 6:

```
oslpcfg disableboot n=6
```

4.3.6.3 Selecting a Boot Device (setbootdevice)

NOTE: This command is deprecated. Use the `setbootparam` command with a boot device parameter to configure a boot device. This section is provided for comparison purposes only.

This command sets the boot device specified by its WWN, LUN, and desired topology.

- Set `t=0` for arbitrated loop.
- Set `t=1` for point-to-point.

The selected device boots when the system reboots.

ATTENTION: Do not select a boot device on an FC adapter that is being used to boot from SAN. The `setbootdevice` command might cause a loss of connectivity to the SAN and possible loss of data. To run the `setbootdevice` 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 run the `setbootdevice` command to a non-boot-from-SAN host, and run the command from there.
- If the host with the `setbootdevice` 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 `setbootdevice` command can now be run on the intended target adapter because it is not being actively used for boot from SAN.

The boot code must be enabled before issuing the `setbootdevice` command.

To set the boot device, type the following command (all on one line):

```
oslpcfg setbootdevice n=<adapter_number> w0=<wwpn_word_0> w1=<wwpn_word_1> l=<Decimal_ID_of_LUN>
t=<topology>
```

NOTE: Enter the LUN in decimal format.

The following example sets the boot device on adapter number 1, LUN number 46, with a desired topology of arbitrated loop:

```
oslpcfg setbootdevice n=1 w0=a1b2c3d4 w1=b946a4e8 l=46 t=0
```

NOTE: If port login fails after 50 ms, the command is retried once.

4.3.6.4 Enabling or Disabling Boot Devices (enablebootdevice or disablebootdevice)

NOTE: These commands are deprecated. Use the `setbootparam` command with the `enablebootfromsan` parameter to enable or disable boot devices. This section is provided for comparison purposes only.

After using the `setbootdevice` command, you can enable or disable the boot device by using the `enablebootdevice` or `disablebootdevice` command.

ATTENTION: Do not enable or disable the boot device on an FC adapter that is being used to boot from SAN. The `enablebootdevice` or `disablebootdevice` command might cause a loss of connectivity to the SAN and possible loss of data. To run either 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 enable or disable the boot device to a non-boot-from-SAN host, and run the command 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. Either command can now be run on the intended target adapter because it is not being actively used for boot from SAN.

To enable the boot device, type the following command:

```
oslpcfg enablebootdevice n=<adapter_number>
```

To disable the boot device, type the following command:

```
oslpcfg disablebootdevice n=<adapter_number>
```

NOTE: For the change to take effect, perform a system reboot.

4.3.6.5 Getting Boot Parameters (getbootparams)

NOTE: This command replaces the deprecated `readaltboot` and `readbootdevice` commands.

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
- `ob`: Open Boot

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
```

4.3.6.6 Reading All Alternate Boot Devices (readaltboot)

NOTE: This command is deprecated. Use the `getbootparams` command to view alternate boot devices. This section is provided for comparison purposes only.

This command shows the WWN and LUN numbers in decimal format of all possible alternate boot devices. You can have up to seven alternate boot devices.

This command also allows you to identify available alternate boot devices that can be configured using the `setaltboot` command.

To read all alternate boot devices, type the following command:

```
oslpcfg readaltboot n=<adapter_number>
```

4.3.6.7 Setting One or More Alternate Boot Devices (setaltboot)

NOTE: This command is deprecated. Use the `setbootparam` command with a boot device parameter to set alternate boot devices. This section is provided for comparison purposes only.

This command configures alternate boot devices. You can set up to seven alternate boot devices; that is, *index* `i` can be from 1 to 7.

NOTE: The boot code must be enabled before you issue the `setaltboot` command.

The `index` parameter in this command does not refer to the same index as in the `enableboot` command.

To set up one or more alternate boot devices, type the following command:

```
oslpcfg setaltboot n=<adapter_number> i=<index> w0=<wwpn_word_0> w1=<wwpn_word_1>
l=<Decimal_ID_of_LUN>
```

The following example sets the alternate boot device on adapter number 1, LUN number 3:

```
oslpcfg setaltboot n=1 i=1 w0=12345678 w1=a842b6ed l=3
```

4.3.7 Configuration Commands

Configuration commands allow you to update the configuration region of an adapter by name or by number.

4.3.7.1 Updating Configuration Regions (config)

Two forms of configuration are available:

- Configure all adapters of a given adapter name at once
- Configure a single adapter by its number

Valid region numbers range from 0 to 32.

4.3.7.1.1 Update by Name

To update a specified configuration region on all adapters of the same selected name, type the following command:

```
oslpcfg config a=<adapter name> r=<region number> c=<configuration filename>
```

NOTE: The adapter name is the name that appears when you run the `listHBA` command. For more information on the `listHBA` command, see Section 4.3.2.7, Viewing All Adapters in the System (listhba).

The following example updates region 6 of all LP12000 adapters with `ctplus1.cfl`:

```
oslpcfg config a=lpe12000 r=6 c=ctplus1.cfl
```

The following example updates region 17 of all ABC24-FC56 adapters with `d:\dfplus1.cfl`:

```
oslpcfg config a=ABC24-FC56 r=17 c=d:\dfplus1.cfl
```

NOTE: The size of the `.cfl` file for the configuration region update can be up to 2028 bytes.

4.3.7.1.2 Update by Number

To update a specified configuration region for one adapter, type the following command:

```
oslpcfg config n=<adapter_number> r=<region_number> c=<configuration_filename>
```

The following example updates region 17 of adapter number 4 with `heplus1.cfl`:

```
oslpcfg config n=4 r=17 c=heplus1.cfl
```

The following example updates region 6 of adapter number 2 with `d:\dfplus1.cfl`:

```
oslpcfg config n=2 r=6 c=d:\dfplus1.cfl
```

4.3.8 Diagnostic Tests

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

4.3.8.1 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
```

4.3.8.2 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.

where:

- `p` = 3 to 8-HEX byte pattern
- `c` = 1500 to 8192
- `r` = 1 to 4096
- `t` = 2

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
```

4.3.8.3 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.

NOTE: The `pciloopback` command supports testing all installed adapters using `n=all` on LPe12000-series adapters only.

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 all LPe12000-series adapters in the system and stops the testing if any errors occur.

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

4.3.8.4 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>
```

4.3.9 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 a=lpe12000 i=c:\temp\ud201a12.all
;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
```

4.3.9.1 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
```

4.3.9.2 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 a=lpe12000 i=c:\temp\ud201a12.all
listboot n=2
setbootparam n=1 efi enableadapterboot 1
```

4.3.9.3 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.

4.3.9.4 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
```

4.3.9.5 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 `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 occurs, 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
```

4.3.9.6 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 `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 `scriptvwwnn` 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
```

4.3.9.7 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: Status Messages and Error Codes

5.1 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` and `boot` directories cannot be located.

Table 4 lists the supported Elxflash utility status messages.

Table 4: 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 or force boot code 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` or `boot` directories are missing. Depending on the operation, autodiscovery expects the `firmware` or `boot` directories to exist as subdirectories 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` or `boot` directories. 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.

5.2 LpCfg Error Codes

Table 5 lists the supported LpCfg error codes.

Table 5: 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.
8	Invalid boot code index.
9	Link not up for external loopback test.
10	Link not up for internal loopback test.
11	Invalid jumper selection (in jumper command).
12	Invalid alternate configuration region (in jumper command).
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.
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.

Table 5: LpCfg Error Codes (Continued)

Error Code	Description
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.
49	POST test 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.
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 customer support for additional help. Download failed on adapter <*adapter number*> Stat 89.

Table 5: LpCfg Error Codes (Continued)

Error Code	Description
90	Download failed due to an invalid firmware digital signature. Contact customer 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.
200	General error.

Chapter 6: Troubleshooting

When using Elxflash, some commands take precedence over others. For instance, the force firmware (`/ff`), force boot (`/fb`), and force firmware and boot code (`/f`) switches take precedence over the `/downgrade`, `/update`, and `/rewrite` switches. If `/ff`, `/fb`, or `/f` 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 /fb /update`
  - `/update` is ignored.
  - Performs a force boot code 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 /fb /downgrade`
  - `/downgrade` is ignored.
  - Performs a force boot code operation on all installed and supported adapters.
- `./elxflash /ff /rewrite`
  - `/rewrite` is ignored.
  - Performs a force firmware operation on all installed and supported adapters.
- `./elxflash /fb /rewrite`
  - `/rewrite` is ignored.
  - Performs a force boot code operation on all installed and supported adapters.

6.1 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.

6.2 Updating Adapters without Boot Code

The Elxflash utility reports an adapter's boot code version as `NONE` when the adapter does not have boot code installed. The utility does not update, downgrade, or rewrite the boot area if boot code is not present, but firmware will still be updated or downgraded without boot code. You can force a boot code download using autodiscovery or the `fwmatrix.txt` file.

BROADCOM®