Solaris FCA Utilities Version 2.90.10.0 User Manual

Emulex, 3333 Susan Street

Costa Mesa, CA 92626

Note: References to OCe11100 series products also apply to OCe11100R series products.

Table of Contents

List of Tables

1. Introduction

Overview

This Solaris Fibre Channel Adapter (FCA) Utilities User Manual provides information for the Emulex emlxadm and emlxdrv utility programs.

- emlxadm – changes driver parameters through a local interactive or command line interface (CLI) mode. It can also update firmware on non-Oracle branded devices.
- emlxdrv – binds (associates) various Fibre Channel (FC) and network interface card (NIC) drivers to various FC and NIC adapter models, respectively.

Supported Drivers

Emulex emlxadm and emlxdrv utility programs support the following FC and NIC drivers:

- emlxs - Solaris inbox Fibre Channel/Fibre Channel over Ethernet (FC/FCoE driver)
- elxfc - Emulex distributed FC/FCoE Solaris driver (does not support Oracle-branded devices)
- lpfc - Legacy Emulex distributed FC SD driver, which does not support Oracle-branded devices. The lpfc driver is available for Solaris 10, but not for Solaris 11.
- oce - Solaris inbox NIC driver
- elxnic - Emulex distributed Solaris NIC driver

Abbreviations

API	application programming interface
CLI	Command line interface
COMSTAR	Common Multiprotocol SCSI Target
D_ID	destination identifier
ELS	Extended Link Service
FC	Fibre Channel
FCA	Fibre Channel adapter
FCIO	FC input/output
FCoE	Fibre Channel over Ethernet
FCP	Fibre Channel Protocol
FCT	Fibre Channel port provider (module in Oracle COMSTAR stack)
FCTL	Fibre Channel transport library

HBA	host bus adapter
iSCSI	internet Small Computer System Interface
LINIT	Loop Initialize ELS command
man	manual (e.g. man pages)
NIC	network interface card (or controller)
PHY	physical layer (module)
RNID	Request Node Identification Data ELS
SAN	storage area network
SFS	SAN Foundation Software
SPARC	Scalable Processor Architecture
UCNA	universal converged networking adapter
VPD	vital product data
WWPN	World Wide Port Name

2. Installing and Removing Utilities

Compatibility

Before installing, select the appropriate Solaris SFS operating system, and then download the Solaris FCA utilities from the Emulex website. The Emulex drivers for Solaris and Solaris FCA utilities support the following operating systems:

- Solaris 10 SPARC
- Solaris 10 for x64 and x86
- Solaris 11 SPARC
- Solaris 11 for x64
- Solaris 11.1 SPARC
- Solaris 11.1 for x64

To determine which adapters are supported by the Solaris SFS FCA drivers (emlxs and elxfc), Solaris Ethernet NIC drivers (oce and elxnic), and the Solaris FCA utilities, see the Emulex website.

Installing the Utilities for Solaris 10

The emlxadm and emlxdrv utilities are bundled into an emlxu utilities package. There are two options for installing the emlxu utilities package:

- Using the emlxu_install script – see "Installing the Utilities Using the emlxu_install Script" in the following section.
- Manually, by using pkgadd – see "Installing the Utilities Manually" on page 11.

Installing the Utilities Using the emlxu_install Script

Note: Although it is possible to install emlxu onto one or more clients from a server, that procedure is not covered in this document; see the Solaris documentation.

Before installing the emlxu utilities package, you must install the Solaris 10 Update 9 (or later) with Oracle-recommended patches. Make sure the emlxs or elxfc driver is v2.60k (or later) and the oce driver is v1.10e (or later).

If an earlier version of the emlxu utilities package is installed, in the following procedure, you are prompted to remove it before the script installs the new version.

To install the emlxu utilities package using the emlxu_install script:

1. Log in as "root", or "su" to root.
2. Copy the emlxu utilities package from your distribution medium into a directory. The emlxu utilities package is a "tar" file, with a name in the following format:

   ```
   emlxu_kit-<version>-sparc.tar
   ```

3. Change to the directory of the tar file:

   ```
   cd <directory>
   ```

4. Extract the emlxu_install script from the tar file:

   ```
   tar xf emlxu_kit-<version>-sparc.tar emlxu_install
   ```

5. Install the emlxu utilities package:

   ```
   emlxu_install
   ```

   If an earlier version of the emlxu utilities package is not found, a message is displayed indicating this, and you can skip to step 7. Otherwise, the script begins removing any earlier version of the emlxu utilities package, and the following message is displayed:

   ```
   <Removing old EMLXemlxu package>
   ```

6. If an earlier version of the emlxu utilities package is installed, you are prompted to remove it:

   ```
   Do you want to remove this package? [y,n,?,q]
   ```

   Enter "y". The following message is displayed:

   ```
   Removal of <EMLXemlxu> was successful.
   ```

7. The script expands the new tar file and begins installing the emlxu utilities package. The following message is displayed:

   ```
   <Expanding emlxu_kit-<version>-sparc.tar>
   <Adding new package>
   ```

   The package is prepared for installation, and you are prompted to confirm its installation:

   ```
   Do you want to continue with the installation of <EMLXemlxu> [y,
   n,?]:
   ```

   Enter "y". The installation progress is indicated.

8. Examine the output for any errors or warnings. If the installation is successful, the following message is displayed near the end of the process:

   ```
   Installation of <EMLXemlxu> was successful.
   ```

   The script performs some cleanup and the following messages are displayed:

   ```
   <Cleaning directory>
   <emlxu_install complete>
   <Execute "emlxu_remove" when ready to uninstall>
   ```

   The script copies the emlxu_remove script into the working directory with the original emlxu utilities package tar file. You can use emlxu_remove script later when you want to remove the emlxu utilities from your system. See "Removing the Utilities for Solaris 10" on page 13 for more details. You can also delete the emlxu_remove script.

   The emlxu utilities installation is complete. The emlxu utilities package's programs are located in the /opt/EMLXemlxu/bin directory.

9. You do not have to reboot the system to run a utility program, but you must either enter the program's full path name, or add the package's bin directory (/opt/EMLXemlxu/bin) to the system environment's search path. To use the man pages provided by the package, you must also add the package's man directory (opt/EMLXemlxu/man) to the system environment's man path.

For more information on installing and removing packages, see the Solaris system administration documentation and the pkgadd(1M) and pkgrm(1M) man pages.

Installing the Utilities Manually

Note: If an earlier version of the EMLXemlxu utilities package is already installed and you want to install a newer version, you must remove the earlier version manually, as detailed in "Removing the Utilities Manually" on page 14, before installing the newer utilities package manually.

To install the utilities package manually:

1. Log in as "root", or "su" to root.
2. Copy the emlxu utilities package from your distribution medium into a directory. The emlxu utilities package is a "tar" file, with a name in the following format:

   ```
   emlxu_kit-<version>-sparc.tar
   ```
3. Go to the directory of the tar file:

   ```
   cd <directory>
   ```
4. Extract the emlxu_install script from the tar file:

   ```
   tar xvf emlxu_kit-<version>-sparc.tar
   ```
5. Install the EMLXemlxu utilities package:

   ```
   pkgadd -d . EMLXemlxu
   ```
6. The package is prepared for installation, and you are prompted to confirm its installation:

   ```
   Do you want to continue with the installation of <EMLXemlxu>
   [y,n,?]
   ```

   Enter "y". The installation progress is indicated.
7. Examine the output for any errors or warnings. If the installation is successful, the following message is displayed near the end of the process:

   ```
   Installation of <EMLXemlxu> was successful.
   ```

   The emlxu utilities installation is complete. The utility package's programs are located in the /opt/EMLXemlxu/bin directory.
8. You do not have to reboot the system to run a utility program, but you must either enter the program's full path name or add the package's bin directory (/opt/EMLXemlxu/bin) to the system environment's search path. To use the man pages provided by the package, you must also add the package's man directory (opt/EMLXemlxu/man) to the system environment's man path.

For more information on installing and removing packages, see the Solaris system administration documentation and the pkgadd(1M) and pkgrm(1M) man pages.

Installing the Utilities for Solaris 11

Note: Do not install the Emulex Solaris FCA Utility Kit (emlxu) from the p5p bundle or IPS repository on any host that has OCM installed. Remove the OCM kit

before installing emlxu using p5p. If you want to run both OCM and emlxu, use the combined installation kit.

Remote Repository Installation

1. Log in as "root", or "su" to root.
2. Add the Emulex IPS repository to the publisher list:

   ```
   $ pkg set-publisher –O http://<repository_url> emulex
   ```

3. List all available versions of emlxu. For example:

   ```
   $ pkg list -af emlxu
   NAME (PUBLISHER   VERSION    IFO
   emlxu (emulex)    1.8.4.0-0  ---
   emlxu (emulex)    1.8.3.0-0  ---
   ```

4. By default, the newest version of emlxu that is compatible with the rest of the image will be installed. On the system whose output was displayed in step 3, the following command will install version 1.8.4.0-0:

   ```
   $ pfexec pkg install emlxu
   ```

   To install a specific version of emlxu, append the package version to the package name as follows:

   ```
   $ pfexec pkg install emlxu@1.8.3.0-0
   ```

   In instances where two publishers provide packages of the same name, specify the publisher name as follows:

   ```
   $ pfexec pkg install //emulex/emlxu
   ```

P5P Archive Installation

Scripted Installation

1. Log in as "root", or "su" to root.
2. Download the p5p tar file from the Emulex website.
3. Untar the archive:

   ```
   $ tar -xvf emlxu_p5p-1.08.4.0.tar
   ```

4. Run the installation script:

   ```
   $ ./emlxu_install
   ```

Manual Installation

1. Log in as "root", or "su" to root.
2. Download the p5p tar file from the Emulex website.
3. Extract the p5p archive from the tar file:

   ```
   $ tar -xvf emlxu_p5p-1.08.4.0.tar
   ```

4. Run the following command to install the package:

   ```
   $ pfexec pkg install –g emlxu_1.08.4.0.p5p emlxu
   ```

5. Reboot the system:

```
$ reboot
```

Removing the Utilities for Solaris 10

There are three options for removing the emlxu utilities package:

- Using the emlxu_install script – removes any previous versions of the emlxu utilities package before installing the latest emlxu utilities package. See "Installing the Utilities Using the emlxu_install Script" on page 9.
- Using the emlxu_remove script – removes all emlxu files. See "Removing the Utilities Using the emlxu_remove Script" in the following section.
- Manually, by using pkgrm – removes all emlxu files. See "Removing the Utilities Manually" on page 14.

Removing the Utilities Using the emlxu_remove Script

To remove all emlxu files using the emlxu_remove script:

1. Log in as "root", or "su" to root.
2. If you are in the directory of the emlxu_remove script, go to step 4. Otherwise, go to the directory where the original emlxu utilities package tar file is located:

```
cd <directory>
```

3. Extract the emlxu_remove script from the emlxu utilities package tar file:

```
tar xf emlxu_kit-<version>-sparc.tar emlxu_remove
```

4. Run the emlxu_remove script:

```
emlxu_remove
```

5. The emlxu_remove script locates the EMLXemlxu package, and the following message is displayed:

```
<Removing EMLXemlxu package>
```

   If an emlxu utilities package is not found, a message is displayed indicating this, and you can skip to step 7, Otherwise, you are prompted to remove the package:

```
Do you want to remove this package? [y,n,?,q]
```

   Enter "y". The following message is displayed:

```
Removal of <EMLXemlxu> was successful.
```

6. The script performs some cleanup, and the following messages are displayed:

```
<Removing emlxu scripts>
<emlxu_remove complete>
```

   The utilities package has been removed.

7. If you want to install another version of the emlxu utilities package, follow the instructions provided in one of the following sections:
   - "Installing the Utilities for Solaris 10" on page 9.
   - "Updating the Utilities" on page 14.

For more information on installing and removing packages, see the Solaris system administration documentation and the pkgadd(1M) and pkgrm(1M) man pages.

Removing the Utilities Manually

To manually remove the EMLXemlxu utilities package:

1. Log in as "root", or "su" to root.
2. Run the package removal command:

   ```
   pkgrm EMLXemlxu
   ```

3. You are prompted to confirm the package removal:

   ```
   Do you want to remove this package? [y,n,?,q]
   ```

   Enter "y". The package is prepared for removal.

4. You are prompted again for confirmation:

   ```
   Do you want to remove this package? [y,n,?,q]
   ```

   Enter "y". The following message is displayed:

   ```
   Removal of <EMLXemlxu> was successful
   ```

For more information on installing and removing packages, see the Solaris system administration documentation and the pkgadd(1M) and pkgrm(1M) man pages.

Removing the Utilities for Solaris 11

Manual Removal

Run the following command to remove emlxu from the system:

```
$ pkg uninstall emlxu
```

Scripted Removal

Run the following command to remove emlxu from the system:

```
$ ./emlxu_remove
```

Updating the Utilities

For Solaris 10

There are two options for updating the utilities package:

- Using the emlxu_install script – follow the procedure in "Installing the Utilities Using the emlxu_install Script" on page 9. In this procedure, if an earlier version of the emlxu utilities package is installed, you are prompted to remove it before installing the newer version.
- Manually – first, manually remove the existing EMLXemlxu utilities package as detailed in "Removing the Utilities Manually" on page 14. Then, manually

install the newer EMLXemlxu utilities package as detailed in "Installing the Utilities Manually" on page 11.

For Solaris 11

There are two options for updating the utilities package:

- Using the remote repository method - run

  ```
  pkg update emlxu
  ```

- Using the p5p installation method - follow either the script or the manual procedure in "P5P Archive Installation" on page 12.

3. Using the emlxadm Utility

The emlxadm utility changes driver parameters through a local interactive or CLI mode. It can also update firmware on non-Oracle branded devices. The emlxadm utility is intended to be a direct user interface to the FCIO interface provided by the Oracle StorEdge SFS. The FCIO interface provides an Oracle common ioctl interface to the FCTL driver, which manages the FCA drivers for each Fibre Channel adapter attached to the host system.

The emlxadm utility program can run in two modes:

- Interactive mode (see "Interactive Mode of Operation for emlxadm" in the following section)
- CLI mode (see "CLI Mode of Operation for emlxadm" on page 19)

Note: The OneCommand Manager application provides the same functions as emlxadm, and a number of additional ones on multiple systems, through a choice of a graphical user interface and a scriptable command line interface.

Interactive Mode of Operation for emlxadm

To run the emlxadm utility in interactive mode, type "emlxadm" without any command line arguments. For example:

```
# emlxadm
```

Displaying Available Emulex Adapters

After the emlxadm utility is started, it scans the host system and prepares a list of qualified adapter ports. Qualified adapter ports are devices that are attached to the emlxs, elxfc, oce, or elxnic driver. After each list number, the display indicates the adapter's type of stack and type of driver. For example:

```
Available Emulex HBA's:

1. SFS:emlxs0   : /devices/pci@0,0/pci10de,5d@c/pci10df,e602@0,2/fp@0,0 (CONNECTED)
2. NIC:oce0     : /devices/pci@0,0/pci10de,5d@c/pci10df,e602@0          (CONNECTED)
3. SFS:emlxs1   : /devices/pci@0,0/pci10de,5d@c/pci10df,e602@0,3/fp@0,0 (CONNECTED)
4. NIC:oce1     : /devices/pci@0,0/pci10de,5d@c/pci10df,e602@0,1        (CONNECTED)
5. FCT:emlxs2   : /devices/pci@0,0/pci10de,5d@d/pci10df,f100@0          (CONNECTED)
6. SFS:emlxs3   : /devices/pci@0,0/pci10de,5d@d/pci10df,f100@0,1/fp@0,0 (CONNECTED)

Enter an HBA number or zero to exit:
```

In this example, six adapter ports are available. For each adapter, the type of stack is indicated. The three types are:

- SFS – the Oracle SFS stack
- FCT – the Oracle COMSTAR stack
- NIC – the Oracle networking stack

After the available adapter list is displayed, you are prompted to choose one of the available adapter ports by entering its list number, or you can type "0" or "zero" to exit.

Selecting an Adapter Port Attached to an SFS or FCT Stack

If you select an adapter port that is attached to an SFS or FCT stack, the emlxadm utility displays a list of available commands. For example, if you enter "1" from the example in "Displaying Available Emulex Adapters" on page 16, the emlxadm utility displays:

```
HBA 1: /devices/pci@0,0/pci10de,5d@c/pci10df,e602@0,2/fp@0,0

Available commands:         [FCIO rev2]

get_num_devs                - Returns the number of FC devices seen by this HBA.
get_dev_list                - Returns a list of FC devices seen by this HBA.
get_logi_params <wwpn>      - Returns the login paramters for a specified FC device.
get_host_params             - Return the host parameters.
get_sym_pname               - Returns the symbolic port name of a device.
set_sym_pname <string>      - Sets the symbolic port name for a device.
get_sym_nname               - Returns the symbolic node name of a device.
set_sym_nname <string>      - Sets the symbolic node name for a device.
dev_login <wwpn>            - Performs an FC login to a device.
dev_logout <wwpn>           - Performs an FC logout to a device.
get_state <wwpn>            - Returns current SFS state of a specified device.
dev_remove <wwpn>           - Remove the FC device from SFS management.
link_status <d_id>          - Request link error status from a specified D_ID.
get_fcode_rev               - Returns the current Fcode revison of the HBA.
download_fcode [filename]   - Download the HBA fcode.
get_fw_rev                  - Returns the current firmware revison of the HBA.
download_fw [filename]      - Download the HBA firmware.
get_boot_rev                - Returns the current boot revison of the HBA.
get_phy_attrs               - Returns the current PHY attributes for the HBA.
download_boot [filename]    - Download the HBA boot image.
get_dump_size               - Returns the HBA's firmware core dump size.
force_dump                  - Force a firmware core dump on this HBA.
get_dump [-h]               - Saves firmware core dump to a file.
get_topology                - Returns the current FC network topology.
reset_link [wwpn]           - Resets the link of a specified public loop FC device.
reset_hard                  - Reset the HBA.
diag ...                    - Perform a diagnostic test on the HBA.
ns                          - Performs a complete query of the fabric name server.
parm_get_num                - Returns the total number of configurable parameters.
parm_get_list               - Returns a list of configurable parameters.
parm_get <label>            - Gets the value of a specified parameter in the driver.
parm_set <label> <val>      - Sets the value of a specified parameter in the driver.
msgbuf [all], <number> [-i interval] - Returns the driver's internal message log.
get_host_attrs              - Returns the host adapter and port attributes.
get_port_attrs <index>, <wwn>, all   - Returns the port attributes.
get_path <index>            - Returns the adapter path.
get_vpd                     - Returns the adapter's Vital Product Data (VPD).
boot_code [enable, disable] - Sets or shows the boot code state in this HBA.
get_rnid [wwpn]             - Gets the RNID information for local or specified port.
```

```
get_inst                    - Gets the driver instance.
get_phy_attrs               - Returns the current PHY attributes for the HBA.(CNA's only)
set_throttle                - Temporarily sets the I/O queue depth for a specified remote
                              port.
get_throttle                - Gets the I/O queue depth for all remote ports.
q                           - Exits this program.
h                           - Returns this help screen.
hba                         - Select another hba.
p                           - Repeat previous command.

emlxadm>
```

Selecting an Adapter Port Attached to a NIC Stack

If you select an adapter port that is attached to a NIC stack, the emlxadm utility presents a list of available commands. For example, if you enter "2" from the example in "Displaying Available Emulex Adapters" on page 16, the emlxadm utility displays:

```
HBA 2: /devices/pci@0,0/pci10de,5d@c/pci10df,e602@0 (physical port)

Available commands:      [NIC rev1]

get_hba_attrs           - Returns the current control attributes for the HBA.
get_linkinfo            - Returns the current link status information for the HBA port.
get_fw_rev              - Returns the current firmware revision of the HBA.
download_fw [filename]  - Download the HBA firmware.
q                       - Exits this program.
h                       - Returns this help screen.
hba                     - Select another HBA.
p                       - Repeat previous command.

emlxadm>
```

Entering emlxadm Commands

After the available commands are listed, the "emlxadm>" prompt is displayed. From this point, the utility is prompt-driven. When the prompt is displayed, you can enter one of the commands in the list. For example, you can display the list of available commands at anytime by typing "h" (the help screen) at the prompt:

```
emlxadm> h
```

For some commands, you may have optional or required arguments. If a command requires an argument, but is entered without the argument, a usage statement is returned to indicate that the command requires an argument(s). For example, the get_state command requires a WWPN for the target device. Therefore, if you type only "get_state":

```
emlxadm> get_state
```

The emlxadm utility returns

```
Usage: get_state <wwpn>
```

Therefore, you must include the <wwpn> argument for the get_state command to receive a valid response. For example:

```
emlxadm> get_state 21000020371938fa
```

The emlxadm utility can now run the command and display the state:

```
State: PORT_DEVICE_LOGGED_IN
```

Exiting emlxadm

To exit (quit) the emlxadm utility, type "q" at the prompt:

```
emlxadm> q
```

CLI Mode of Operation for emlxadm

There are two options available to run emlxadm in a CLI mode:

- Device path option (see the following section)
- Instance option (see page 21)

Device Path Option in CLI Mode

In this CLI mode option, the user types "emlxadm", the device path, followed by a valid command, and its command arguments (if applicable).

Syntax

```
emlxadm <device path> [-y] <cmd> [cmd_option(s)]
```

Arguments

`device path`	Specifies the full device name for a single adapter or a pattern string for multiple adapters. If the pattern string matches any part of an adapter device path, the command runs on that adapter.
`-y`	When the [-y] option is included, the emlxadm utility runs immediately without pausing for a verification from the user to continue. When the [-y] option is absent, the emlxadm utility pauses for a verification from the user before running the command.
`cmd`	An emlxadm command. See Table 3-1, Summary for emlxadm Commands, on page 23.
`cmd_option(s)`	Various emlxadm command arguments, if applicable.

Device Path Option for a Single Adapter

When using the device path option for single adapters, the <device path> parameter must be the full device name for the single adapter.

Running a Command with User Verification

In this example, the emlxadm utility pauses for a verification from the user before running the command. The user types "emlxadm", the full device name (`/devices/pci@1e,600000/SUNW,emlxs@2/fp@0,0:devctl`), the command (`get_state`), and its <wwpn> argument (`21000020371938fa`):

```
# emlxadm /devices/pci@1e,600000/SUNW,emlxs@2/fp@0,0:devctl
get_state 21000020371938fa
```

The elxadm utility returns some status, but pauses for a response before running the command:

```
Found path to 1 HBA port(s).
HBA port:  /devices/pci@1e,6000000/SUNW,emlxs@2/fp@0,0:devctl
>Do you wish to continue with this device [y,n,q] ? y
```

After the user types "y", the emlxadm utility runs the command and displays the result:

```
State: PORT_DEVICE_LOGGED_IN
#
```

Running a Command without User Verification (Using [-y])

If you do not want the emlxadm utility to pause for verification before running the command, include the "-y" option after the full device name. For example, when the user types "emlxadm", the "-y" option, the full device name, the command, and its argument, the elxadm utility runs the command immediately:

```
# emlxadm /devices/pci@1e,600000/SUNW,emlxs@2/fp@0,0:devctl  -y
get_state  21000020371938fa

Found path to 1 HBA port(s).
HBA port:  /devices/pci@1e,600000/SUNW,emlxs@2/fp@0,0:devctl
State: PORT_DEVICE_LOGGED_IN
#
```

Device Path Option for Multiple Adapters

When using the device path option for multiple adapters, use a pattern string for the <device path> parameter. If the pattern string matches any part of an adapter device path, the command runs on that adapter.

Running a Command with User Verification

In this example, the emlxadm utility pauses for a verification from the user before running the command. The user types "emlxadm", the pattern string ("SUNW,emlxs@2"), and the command (`get_num_devs`):

```
# emlxadm  "SUNW,emlxs@2"  get_num_devs
```

The elxadm utility returns some status, but pauses for a response before running the command:

```
Found path to 2 HBA port(s).
HBA port:  /devices/pci@1e,600000/SUNW,emlxs@2/fp@0,0:devctl
> Do you wish to continue with this device [y,n,q] ?  y
```

After the user types "y", the elxadm utility returns more information, but again pauses for a response before running the command:

```
There are 5 devices reported on this port.
HBA port:  /devices/pci@1e,600000/SUNW,emlxs@2,1/fp@0,0:devctl
> Do you wish to continue with this device [y,n,q] ?  y
```

After the user types "y", the emlxadm utility runs the command and displays the result:

```
There are 0 devices reported on this port.
#
```

Running a Command without User Verification (Using [-y])

If you do not want the emlxadm utility to pause for verification before running the command, include the "-y" option after the pattern string. For example, when the user types "emlxadm", the "-y" option, the pattern string, and the command, the elxadm utility runs the command immediately:

```
# emlxadm  "SUNW,emlxs@2"  -y  get_num_devs

Found path to 2 HBA port(s).

HBA port:  /devices/pci@1e,600000/SUNW,emlxs@2/fp@0,0:devctl
There are 5 devices reported on this port.

HBA port:  /devices/pci@1e,600000/SUNW,emlxs@2,1/fp@0,0:devctl
There are 0 devices reported on this port.
#
```

Instance Option in CLI Mode

This CLI mode option enables you to use the emlxadm utility as part of a script or another program capable of running system-level calls.

Syntax

```
emlxadm -i<N,SFS,FCT or all> [-y] <cmd> [cmd_option(s)]
```

Arguments

`N`	Indicates a specific emlxs driver instance. For example, N=1 means "emlxs1", N=2 means "emlxs2", and N=2.1 means "emlxs2.1".
`SFS`	Indicates all emlxs driver instances that are attached to the Oracle SFS interface.
`FCT`	Indicates all emlxs driver instances that are attached to the Oracle COMSTAR interface.
`all`	Indicates all emlxs driver instances.
`-y`	When the [-y] option is included, the emlxadm utility runs immediately. It does not pause for a verification from the user to continue. When the [-y] option is absent, the emlxadm utility pauses for a verification from the user before running the command.
`cmd`	An emlxadm command. See Table 3-1, Summary for emlxadm Commands, on page 23.
`cmd_option(s)`	Various emlxadm command arguments, if applicable.

Using "emlxadm help" for Command Usage in CLI Mode

The emlxadm utility offers help for command usage in CLI mode. To invoke a usage help screen, type "`emlxadm help`" at the prompt. The CLI mode usage screen is displayed as follows:

```
USAGE:   emlxadm      :Runs utility in interactive mode.
         or
         emlxadm -v
         or
         emlxadm -i<N, SFS, FCT, or all> [-y] <cmd> [cmd_option(s)]
         or
         emlxadm -j<N, SFS, FCT, or all> [-y] <cmd> [cmd_option(s)]
         or
         emlxadm -n<N or all> [-y] <cmd> [cmd_option(s)]
         or
         emlxadm -m<N or all> [-y] <cmd> [cmd_option(s)]
         or
         emlxadm <device_path> [-y] <cmd> [cmd_option(s)]
OPTIONS:
         -v           Display utility version information.
         -i<N>        Executes command on a specific emlxs driver
                      instance.(Example: N=2 for emlxs2 or N=2.1 for emlxs2.1)
         -iSFS        Executes command on all SFS emlxs driver instances.
         -iFCT        Executes command on all FCT emlxs driver instances.
         -iall        Executes command on all emlxs driver instances.
```

-j<N>	Executes command on a specific elxfc driver instance.(Example: N=2 for elxfc2 or N=2.1 for elxfc2.1)
-jSFS	Executes command on all SFS elxfc driver instances.
-jFCT	Executes command on all FCT elxfc driver instances.
-jall	Executes command on all elxfc driver instances.
-n<N>	Executes command on a specific oce driver instance.(Example: N=2 for oce2 or N=5 for oce5)
-nall	Executes command on all oce driver instances.
-m<N>	Executes command on a specific elxnic driver instance.(Example: N=2 for elxnic2 or N=5 for elxnic5)
-mall	Executes command on all oce driver instances.
device_path	If a full device path is not specified, then the command will be executed on all device paths containing the specified device_path string.
-y	If multiple devices are found, the utility will ask for verification before executing the command on each device. This option will cause the utility to skip the verification and automatically execute the command on each device.

Command Descriptions for emlxadm

This section provides a list of commands and descriptions that can be issued with the emlxadm utility. Table 3-1 summarizes this command list, including abbreviated descriptions.

Table 3-1 Summary for emlxadm Commands

Command Syntax	Description	Page
boot_code [enable, disable]	Sets or shows the boot code state of the current adapter.	27
dev_login <wwpn>	Performs an FC login to an FC device on the network, if not already logged in.	27
dev_logout <wwpn>	Performs an FC logout to an FC device on the network, if not already logged out.	27
dev_remove <wwpn>	Removes the specified FC device from Solaris SFS management. **Caution:** This command is not supported in the Solaris SFS stack and causes the host operating system to panic.	27

Table 3-1 Summary for emlxadm Commands (Continued)

Command Syntax	Description	Page

`diag...`	Performs a diagnostic test on the adapter. Two formats are available:
	`diag <test [parameters]>`
	Performs a diagnostic test on the adapter port specified by an Emulex-specific test and its parameters (if applicable).
	`diag code <cmd_code>`
	Performs a diagnostic test on the adapter port specified by a diagnostic command code (in hexadecimal). This command provides generic support to issue an adapter-specific diagnostic code to any third-party adapter.
		27
		
		28

`download_boot <filename>`	Downloads the specified boot image file to the adapter.	28
`download_fcode <filename>`	Downloads the specified FCode image file to the adapter.	29
`download_fw <filename>`	Downloads the specified firmware image file to the adapter.	29
`force_dump`	Forces a firmware core dump on the adapter.	30
`get_boot_rev`	Returns the current boot revision of the adapter.	30
`get_dev_list`	Returns a list of FC devices currently seen by this adapter port.	30
`get_dump [-h]`	Saves the firmware core dump to a file.	32
`get_dump_size`	Returns the adapter's firmware core dump size.	32
`get_fcode_rev`	Returns the current FCode revision of the adapter.	32
`get_fw_rev`	Returns the current firmware revision of the adapter.	32
`get_hba_attrs`	Returns the current control attributes for the adapter.	32
`get_host_attrs`	Displays all of the current host adapter API attributes.	33
`get_host_params`	Returns the FC login parameters of this adapter port.	34
`get_inst`	Returns driver instance for this adapter port.	35
`get_linkinfo`	Returns the current link status information for the adapter port.	35
`get_logi_params <wwpn>`	Returns the FC login common service parameters for a specified FC device on the network.	35
`get_num_devs`	Returns the number of FC devices currently seen by this adapter port.	29
`get_path <index>`	Shows the current Solaris device path for a specified adapter port. The total number of ports available can be seen in the “Number of HBA ports” attribute displayed using the get_host_attrs command. The <index> argument is an index into this list.	36

Table 3-1 Summary for emlxadm Commands (Continued)

Command Syntax	Description	Page
get_phy_attrs	Shows the current PHY module attributes of the adapter. **Note:** This command applies to UCNAs only.	37
get_port_attrs <index>, <wwn>, all	Shows the current adapter API port attributes. All of the ports' attributes can be displayed, or a single port can be specified by <index> or <wwn>. The total number of ports available can be seen in the "Number of Discovered Ports" attribute displayed using the get_host_attrs command. The <index> argument is an index into this list.	37
get_rnid [wwpn]	Returns the RNID information for the local or specified port.	39
get_state <wwpn>	Returns the current Solaris SFS state of the specified FC device on the network.	40
get_sym_nname	Returns the symbolic FC node name of the adapter port. **Note:** This operation is not supported by the Solaris SFS stack.	40
get_sym_pname	Returns the symbolic FC port name of the adapter port. **Note:** This operation is not supported by the Solaris SFS stack.	40
get_throttle	Returns the I/O queue depth of all remote ports.	40
get_topology	Returns the FC network topology of the adapter port.	41
get_vpd	Shows the current adapter's vital product data.	41
h	Returns the help screen, that is, it lists the available commands.	41
hba	Allows you to select another adapter with which to interface. This prevents you from having to exit and reenter the program.	42
link_status <d_id>	Requests and returns the current link error status from the FC device specified by the D_ID address.	43
msgbuf [all], <number> [-i interval]	This command displays the current driver log, with various options.	43
ns	Performs and returns a complete query of the fabric name server.	44
p	Repeats the last command.	45
parm_get <label>	Retrieves the value of a specified parameter in the driver.	45
parm_get_list	Returns a list of configurable parameters.	46
parm_get_num	Returns the total number of configurable parameters.	45
parm_set <label> <value>	Sets the value of a specified parameter in the driver. Only dynamic parameters can be set.	49
q	Exits (quits) the utility program.	49
reset_hard	Forces the adapter to perform a hardware reset.	50

Table 3-1 Summary for emlxadm Commands (Continued)

Command Syntax	Description	Page
`reset_link <[wwpn] or 0>`	If the [wwpn] parameter is specified, this command resets the link of the specified FC device on the network. If "0"(zero) is specified, this command resets the local link.	50
`set_sym_nname <"string">`	Sets the symbolic FC node name of the adapter to the string provided. **Note:** This operation is not supported by the Solaris SFS stack.	50
`set_sym_pname <"string">`	Sets the symbolic FC port name of the adapter to the string provided. **Note:** This operation is not supported by the Solaris SFS stack.	50
`set_throttle`	Temporarily sets the I/O queue depth for a specified remote port.	51

boot_code [enable, disable]

Shows or sets the boot code state of the current adapter.

Examples

To show the current boot code:

```
emlxadm> boot_code
Boot code: Disabled
```

To enable the boot code:

```
emlxadm> boot_code enable
Boot code: Enabled
```

To disable the boot code:

```
emlxadm> boot_code disable
Boot code: Disabled
```

dev_login <wwpn>

Performs an FC login to an FC device on the network, if it is not already logged in.

Example

```
emlxadm> dev_login 21000020371938fa
Done.
```

dev_logout <wwpn>

Performs an FC logout to an FC device on the network, if not already logged out.

Example

```
emlxadm> dev_logout 21000020371938fa
Done.
```

dev_remove <wwpn>

Removes the specified FC device from Solaris SFS management.

Caution: This command is not supported in the Solaris SFS stack and causes the host operating system to panic.

diag <test [parameters]>

Performs the diagnostics function on the adapter port. This command provides support for Emulex-specific tests.

<test [parameters]>

`emlx_biu [pattern]`	Performs the bus interface unit test. The [pattern] parameter is a 4-byte hexadecimal pattern to be used for the test (for example, 0xA5A5A5A5).
`emlx_echo <did> [pattern]`	Performs the echo test to a specified port id. The [pattern] parameter is a 4-byte hexadecimal pattern to be used for the test (for example, 0xA5A5A5A5).
`emlx_post`	Performs the power-on self tests.

Examples

Performs the bus interface unit test:

```
emlxadm> diag emlx_biu
Result: EMLX_DIAG_BIU: Operation successful.
```

Performs the echo test to a <did>=fffffc:

```
emlxadm> diag emlx_echo fffffc
Result: EMLX_DIAG_ECHO: Operation successful.
```

Performs the power-on self tests:

```
emlxadm> diag emlx_post
Result: EMLX_DIAG_POST: Operation successful.
```

diag code <cmd_code>

Performs a diagnostic test on the adapter port specified by a diagnostic command code (in hexadecimal format). This command provides generic support to issue an adapter-specific diagnostic code to any third-party adapter.

Note: The return status from the adapter is displayed in decimal and hexadecimal format if the diagnostic command code is valid for the adapter. No interpretation of the return status is provided.

Examples

```
emlxadm> diag code 0x4526
Result: CODE(0x4526): 16 (0x10)
```

download_boot <filename>

Downloads the specified boot image file to the adapter.

Example

```
emlxadm> download_boot TD190A4.PRG

Image Components: REL type   size=143416
    DWC file:       BOOT: version=03845054, 1.90a4
```

```
Current: Boot: 1.90a3
New:     Boot: 1.90a4   143416 (0x23038) bytes

Are you sure you want to download this image? (y or n): y
Downloading...
Done.
```

download_fcode <filename>

Downloads the specified FCode image file to the adapter.

Example

```
emlxadm> download_fcode LP10000DC-S.fcode
Image Components: REL type   size=33848
    DWC file:       BOOT: version=03841512, 1.50a2

Current: Fcode: 1.50a2
New:     Fcode: 1.05e   33848 (0x8438) bytes

Are you sure you want to download this image? (y or n): y

Downloading...
Result: Operation successful.
Done.
```

download_fw <filename>

Downloads the specified firmware image file to the adapter.

Note: The Emulex FCA driver does not allow a firmware update to Oracle-branded adapters.

Example

```
emlxadm> download_fw LP10000DC-S.fw
Image Components: NOP type
    AWC file:       KERN: version=ff801315, 1.30a5
    DWC file:       SLI2: version=07831914, 1.90a4
    DWC prog:       TEST: version=00f51010, 1.00a0
    DWC prog:       STUB: version=02881914, 1.90a4
    DWC prog:       SLI1: version=06831914, 1.90a4
    DWC prog:       SLI2: version=07831914, 1.90a4

Current: Firmware: 1.90a4
```

```
New:     Firmware: 1.05e 366712 (0x59878) bytes

Are you sure you want to download this image? (y or n): y

Downloading...
Done.
```

force_dump

Forces a firmware core dump on the adapter.

Example

```
emlxadm> force_dump
Done.
```

get_boot_rev

Returns the current boot revision of the adapter.

Example

```
emlxadm> get_boot_rev
Firmware revision: LP10000DC-S 1.90a3
```

get_dev_list

Returns a list of FC devices currently seen by this adapter port.

Example

```
emlxadm> get_dev_list
----------------------------------------------
Device 0:
          Dtype: 0
FC4_type[proto]: 0x00000100, 0x00000000, 0x00000000, 0x00000000,
                 0x00000000, 0x00000000, 0x00000000, 0x00000000
          State: Logged_In
           D_id: 113e1
           LILP: 0
      Hard Addr: e1
           WWPN: 21000020371938fa
           WWNN: 20000020371938fa
----------------------------------------------
Device 1:
          Dtype: 0
```

```
FC4_type[proto]: 0x00000100, 0x00000000, 0x00000000, 0x00000000,
                 0x00000000, 0x00000000, 0x00000000, 0x00000000
          State: Logged_In
           D_id: 113e2
           LILP: 0
      Hard Addr: e2
           WWPN: 21000020371939a2
           WWNN: 20000020371939a2
-----------------------------------------------
Device 2:
          Dtype: 0
FC4_type[proto]: 0x00000100, 0x00000000, 0x00000000, 0x00000000,
                 0x00000000, 0x00000000, 0x00000000, 0x00000000
          State: Logged_In
           D_id: 113e4
           LILP: 0
      Hard Addr: e4
           WWPN: 21000020371938a3
           WWNN: 20000020371938a3
-----------------------------------------------
Device 3:
          Dtype: 0
FC4_type[proto]: 0x00000100, 0x00000000, 0x00000000, 0x00000000,
                 0x00000000, 0x00000000, 0x00000000, 0x00000000
          State: Logged_In
           D_id: 113e8
           LILP: 0
      Hard Addr: e8
           WWPN: 2100002037193670
           WWNN: 2000002037193670
```

get_dump [-h]

Saves the firmware core dump to a file.

Example

Result if a valid dump exists in the driver memory:

```
emlxadm> get_dump -h
   Core size: 6580624 bytes
       files: 2
    TXT file: 13728
    DMP file: 6566876
```

Result if a valid dump does not exist in the driver memory:

```
emlxadm> get_dump -h
No core file available.
```

get_dump_size

Returns the adapter's firmware core dump size.

Example

Result if a valid dump exists in the driver memory:

```
emlxadm> get_dump_size
Size: 6580624 (0x646990) bytes
```

Result if a valid dump does not exist in the driver memory:

```
emlxadm> get_dump_size
Size: 0 (0x0) bytes
```

get_fcode_rev

Returns the current FCode revision of the adapter.

Example

```
emlxadm> get_fcode_rev
FCODE revision: LP10000-S 1.41a3
```

get_fw_rev

Returns the current firmware revision of the adapter.

Example

```
emlxadm> get_fw_rev
Firmware revision: LP10000DC-S 1.90a3
```

get_hba_attrs

Returns the current control attributes for the adapter.

Example

```
emlxadm> get_hba_attrs

HBA:

     Flash ROM Version: SE HBA ATTR VER:0000.0001
          Manufacturer: Emulex Corporation
        Support Modes: TOE,NIC,VM,FCOE_INI,LRO,0
      SEEPROM Version: 2.32
        IOCTL Version: 0x00012345
        EP Fw Version: 0x00897654
```

```
          NCSI Version:
  Def Extended Timeout: 0
                 Model: OneConnect OCe10100
           Description: BladeEngine 2 10Gb FCoE Initiator
         Serial Number: 0x42543934313832373831000000000000000000000000000000
            IP Version: IPv4
      Firmware Version: 2.701.374.2
          BIOS Version: 2.701.374.2
       REDBOOT Version: 2.0.38.0
        Driver Version:
Flash Firmware Version: 2.701.374.2
   Functions Supported: 0
        Max CDB Length: 0
         ASIC Revision: A2
     Generational GUID:
        HBA Port Count: 2
  Multifunction Device: True
         iSCSI Version: 11
   Def Linkdown Timout: 0
         Physical Port: 0
 Max Domains Supported: 1
            HBA Status: Operational
           Cache Valid: True
               HBA MTU: 8342 8342 0 0 0 0 0 0
  Firmware Post Status: 0xC000
       iSCSI Features: LIST_MODE,0
        PCI Vendor ID: 0x19a2
        PCI Device ID: 0x0700
     PCI SubVendor ID: 0x10df
     PCI SubDevice ID: 0xe602
       PCI Bus Number: 0
    PCI Device Number: 0
   PCI Function Number: 0
        Interface Type: 0
    Unique Identifier: 0
          Net Filters: 8
```

get_host_attrs

Displays all of the current host adapter API attributes.

Example

```
emlxadm> get_host_attrs

Host Attributes:
   Manufacturer              = Oracle Microsystems, Inc.
   Serial Number             = BG43918495
   Model                     = LP10000DC-S
```

```
   Model Description          = EMULEX LIGHTPULSE LP10000DC-S 2GB PCI-X
                                FIBRE CHANNEL ADAPTER
   Node WWN                   = 20000000C942097E
   Node Symbolic Name         = none
   Hardware Version           = 1001206d
   Driver Version             = 1.11f.t3 (2006.04.25.11.43)
   Optional ROM Version       = 1.50a9test1
   Firmware Version           = 1.91b5
   Vendor Specific ID         = fc00
   Number of HBA ports        = 1
   Driver Name                = Emulex-S s9-64 sparc v1.11f.t3
   Last Change                = 5
   fp Instance                = e
   Node WWN                   = 20000000C942097E
   Port WWN                   = 10000000C942097E
   Port Fc Id                 = 011700
   Port Type                  = Nport
   Port State                 = Online
   Port Supported COS         = Class3
   Port Supported FC4 Types   = 00000000, 00000000, 00000000, 00000000,
                                00000000, 00000000, 00000000, 00000000,
   Port Active FC4 Types      = 00000120, 00000000, 00000000, 00000000,
                                00000000, 00000000, 00000000, 00000000,
   Port Symbolic Name         = none
   Port Supported Speed       = 1Gb, 2Gb
   Port Speed                 = 1Gb
   Port Max Frame Size        = 0x800 bytes
   Fabric Name                = 0000000000000000
   Number of Discovered Ports = 4
```

get_host_params

Returns the FC login parameters of this adapter port.

Example

```
emlxadm> get_host_params
Host:
          Dtype: 0
FC4_type[proto]: 0x00000120, 0x00000000, 0x00000000, 0x00000000,
                 0x00000000, 0x00000000, 0x00000000, 0x00000000
          State: Online
      Linkspeed: 1Gb
           D_id: 11700
           LILP: 5
      Hard Addr: 0
           WWPN: 10000000c942097e
           WWNN: 20000000c942097e
```

get_inst

Returns driver instance for this adapter port.

Example

```
emlxadm> get_inst
emlxs0
```

get_linkinfo

Returns the current link status information for the adapter port.

Example

```
emlxadm> get_linkinfo

      Physical port: 0
         MAC Duplex: Full Duplex
          MAC Speed: 10 Gbps
          MAC Fault: None
    Mgmt MAC Duplex: None
     Mgmt MAC Speed: None
     QOS Link Speed: Disabled
Logical Link Status: Link Up
```

get_logi_params <wwpn>

Returns the FC login common service parameters for a specified FC device on the network.

Example

```
emlxadm> get_logi_params 21000020371938fa
Login Parameters:
00 00 00 00
20 20 00 00
88 00 08 00
00 ff 00 02
00 00 01 f4
21 00 00 20
37 19 38 fa
20 00 00 20
37 19 38 fa
00 00 00 00
00 00 00 00
00 00 00 00
```

```
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
80 00 00 00
00 00 08 00
00 ff 00 00
00 01 00 00
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
```

get_num_devs

Returns the number of FC devices currently seen by this adapter port.

Example

```
emlxadm>   get_num_devs
There are 4 devices reported on this port.
```

get_path <index>

Shows the current Solaris device path for a specified adapter port. The total number of ports available can be seen in the "Number of HBA ports" attribute displayed using the get_host_attrs command (page 33). The <index> argument is an index into this list.

Example

```
emlxadm> get_path 0
Adapter: /pci@1e,600000/SUNW,emlxs@2/fp@0,0

emlxadm> get_path 1
Adapter: /pci@1e,600000/SUNW,emlxs@2,1/fp@0,0
```

get_phy_attrs

Shows the current physical layer module attributes of the HBA. This command applies to UCNAs only.

Example

```
emlxadm> get_phy_attrs

PHY Attributes:
   PHY Type: XAUI
   Interface Type: CX4 10 GB
   Flags: 0x00000000
```

get_port_attrs <index>, <wwn>, all

Shows the current adapter API port attributes. All of the ports' attributes can be displayed, or a single port can be specified by <index> or <wwn>. The total number of ports available can be seen in the "Number of Discovered Ports" attribute displayed using the get_host_attrs command. The <index> argument is an index into this list.

Example

```
emlxadm> get_port_attrs all

Host Port Attributes:

  Last Change               = 5
  fp Instance               = e
  Node WWN                  = 20000000C942097E
  Port WWN                  = 10000000C942097E
  Port Fc Id                = 011700
  Port Type                 = Nport
  Port State                = Online
  Port Supported COS        = Class3
  Port Supported FC4 Types  = 00000000, 00000000, 00000000, 00000000,
                              00000000, 00000000, 00000000, 00000000
  Port Active FC4 Types     = 00000000, 00000000, 00000000, 00000000,
                              00000000, 00000000, 00000000, 00000000
  Port Symbolic Name        = none
  Port Supported Speed      = 1Gb, 2Gb
  Port Speed                = 1Gb
  Port Max Frame Size       = 0x800 bytes
  Fabric Name               = 0000000000000000
  Number of Discovered Ports = 4
```

```
Port[0] Attributes:

  Node WWN                 = 20000020371938FA
  Port WWN                 = 21000020371938FA
  Port Fc Id               = 0113e1
  Port Type                = Unknown
  Port State               = Unknown
  Port Supported COS       = Class3
  Port Supported FC4 Types = 00000000, 00000000, 00000000, 00000000,
                             00000000, 00000000, 00000000, 00000000
  Port Active FC4 Types:   = 00000000, 00000000, 00000000, 00000000,
                             00000000, 00000000, 00000000, 00000000
  Port Symbolic Name       = SEAGATE ST39103FC       0004
  Port Supported Speed     = Unknown
  Port Speed               = Unknown
  Port Max Frame Size      = 0x0 bytes
  Fabric Name              = 0000000000000000

Port[1] Attributes:

  Node WWN                   = 20000020371938A2
  Port WWN                   = 21000020371938A2
  Port Fc Id                 = 0113e2
  Port Type                  = Unknown
  Port State                 = Unknown
  Port Supported COS         = Class3
  Port Supported FC4 Types:  = 00000000, 00000000, 00000000, 00000000,
                               00000000, 00000000, 00000000, 00000000
  Port Active FC4 Types:     = 00000000, 00000000, 00000000, 00000000,
                               00000000, 00000000, 00000000, 00000000
  Port Symbolic Name         = SEAGATE ST39103FC       0004
  Port Supported Speed       = Unknown
  Port Speed                 = Unknown
  Port Max Frame Size        = 0x0 bytes
  Fabric Name                = 0000000000000000

Port[2] Attributes:

  Node WWN                   = 20000020371939A3
  Port WWN                   = 21000020371939A3
  Port Fc Id                 = 0113e4
  Port Type                  = Unknown
  Port State                 = Unknown
```

```
  Port Supported COS          = Class3
  Port Supported FC4 Types:   = 00000000, 00000000, 00000000, 00000000,
                                00000000, 00000000, 00000000, 00000000
  Port Active FC4 Types:      = 00000000, 00000000, 00000000, 00000000,
                                00000000, 00000000, 00000000, 00000000
  Port Symbolic Name          = SEAGATE ST39103FC       0004
  Port Supported Speed        = Unknown
  Port Speed                  = Unknown
  Port Max Frame Size         = 0x0 bytes
  Fabric Name                 = 0000000000000000

Port[3] Attributes:

  Node WWN                    = 2000002037193670
  Port WWN                    = 2100002037193670
  Port Fc Id                  = 0113e8
  Port Type                   = Unknown
  Port State                  = Unknown
  Port Supported COS          = Class3
  Port Supported FC4 Types:   = 00000000, 00000000, 00000000, 00000000,
                                00000000, 00000000, 00000000, 00000000
  Port Active FC4 Types:      = 00000000, 00000000, 00000000, 00000000,
                                00000000, 00000000, 00000000, 00000000
  Port Symbolic Name          = SEAGATE ST39103FC       0004
  Port Supported Speed        = Unknown
  Port Speed                  = Unknown
  Port Max Frame Size         = 0x0 bytes
  Fabric Name                 = 0000000000000000
```

get_rnid [wwpn]

Returns the Request Node Identification Data ELS information for the local or specified port.

Example

```
emlxadm> get_rnid
Offset: 00 01 02 03  04 05 06 07   08 09 0A 0B   0C 0D 0E 0F   ASCII:
-------------------------------------------------------------------------------
 0:     31 30 30 30  30 30 30 30   63 39 34 63   62 64 34 65   10000000c94cbd4e
10:     00 00 00 07  00 00 00 00   00 00 00 00   00 01 00 00   ................
20:     00 00 00 00  00 00 00 00   00 00 00 00   00 00 00 00   ................
30:     00 00 00 00                                           ....
Done.
```

get_state <wwpn>

Returns the current Solaris SFS state of the specified FC device on the network.

Example

```
emlxadm> get_state 21000020371938fa

State: PORT_DEVICE_LOGGED_IN
```

get_sym_nname

Returns the symbolic FC node name of the adapter port.

Note: This operation is not supported by the Solaris SFS stack.

Example

```
emlxadm> get_sym_nname
ioctl: FCIO_GET_SYM_NNAME: Operation not supported
```

get_sym_pname

Returns the symbolic FC port name of the adapter port.

Note: This operation is not supported by the Solaris SFS stack.

Example

```
emlxadm> get_sym_pname
ioctl: FCIO_GET_SYM_PNAME: Operation not supported
```

get_throttle

Returns the I/O queue depth of all remote ports. The queue depth represents the maximum concurrent I/Os the driver allows to the remote port at any given time. The default queue depth for an FCP target port is specified by the "target-depth" driver parameter.

Example

```
emlxadm> get_throttle

    ____________WWPN: Depth
    21000011c6810947: 512
    21000011c681065f: 512
    21000011c68108c0: 512
    21000011c681061d: 512
```

get_topology

Returns the FC network topology of the adapter port.

Example

```
emlxadm> get_topology
Topology: PRIVATE_LOOP
```

get_vpd

Displays the current adapter's vital product data.

Example

```
emlxadm> get_vpd
Vital Product Data:
     Identifier (ID): FC2G PCI-X LP10000DC - Oracle
    Part Number (PN): LP10000DC-S
   Manufacturer (MN): Oracle.
  Serial Number (SN): BG43918495
    Description (V1): EMULEX LIGHTPULSE LP10000DC-S 2GB PCI-X FIBRE CHANNEL ADAPTER
          Model (V2): LP10000DC-S
  Program Types (V3): T2:83,88,T3:84,T6:83,T7:83,TB:83,TFF:80
    Port Number (V4): 0
```

h

Returns the help screen of the available commands.

Example

This example shows the help screen for an adapter port that is attached to an SFS or FCT stack.

```
emlxadm> h

Available commands:         [FCIO rev2]

get_num_devs                - Returns the number of FC devices seen by this HBA.
get_dev_list                - Returns a list of FC devices seen by this HBA.
get_logi_params <wwpn>      - Returns the login paramters for a specified FC device.
get_host_params             - Return the host parameters.
get_sym_pname               - Returns the symbolic port name of a device.
set_sym_pname <string>      - Sets the symbolic port name for a device.
get_sym_nname               - Returns the symbolic node name of a device.
set_sym_nname <string>      - Sets the symbolic node name for a device.
dev_login <wwpn>            - Performs an FC login to a device.
dev_logout <wwpn>           - Performs an FC logout to a device.
get_state <wwpn>            - Returns current SFS state of a specified device.
dev_remove <wwpn>           - Remove the FC device from SFS management.
link_status <d_id>          - Request link error status from a specified D_ID.
get_fcode_rev               - Returns the current Fcode revison of the HBA.
```

```
download_fcode [filename]   - Download the HBA fcode.
get_fw_rev                  - Returns the current firmware revison of the HBA.
download_fw [filename]      - Download the HBA firmware.
get_boot_rev                - Returns the current boot revison of the HBA.
download_boot [filename]    - Download the HBA boot image.
get_dump_size               - Returns the HBA's firmware core dump size.
force_dump                  - Force a firmware core dump on this HBA.
get_dump [-h]               - Saves firmware core dump to a file.
get_topology                - Returns the current FC network topology.
reset_link [wwpn]           - Resets the link of a specified public loop FC device.
reset_hard                  - Reset the HBA.
diag ...                    - Perform a diagnostic test on the HBA.
ns                          - Performs a complete query of the fabric name server.
parm_get_num                - Returns the total number of configurable parameters.
parm_get_list               - Returns a list of configurable parameters.
parm_get <label>            - Gets the value of a specified parameter in the driver.
parm_set <label> <val>      - Sets the value of a specified parameter in the driver.
msgbuf [all], <number> [-i interval] - Returns the driver's internal message log.
get_host_attrs              - Returns the host adapter and port attributes.
get_port_attrs <index>, <wwn>, all   - Returns the port attributes.
get_path <index>            - Returns the adapter path.
get_vpd                     - Returns the adapter's Vital Product Data (VPD).
boot_code [enable, disable] - Sets or shows the boot code state in this HBA.
get_rnid [wwpn]             - Gets the RNID information for local or specified port.
get_inst                    - Gets the driver instance.
get_phy_attrs               - Returns the current PHY attributes for the HBA.(CNA's only)
set_throttle...             - Temporarily sets the I/O queue depth for a specified remote
                              port.
get_throttle                - Gets the I/O queue depth for all remote ports.
q                           - Exits this program.
h                           - Returns this help screen.
hba                         - Select another hba.
p                           - Repeat previous command.

emlxadm>
```

hba

Selects another adapter with which to interface. This command allows you to connect to another adapter without having to exit and re-enter the program.

Example

```
emlxadm> hba

Available Emulex HBA's:

1. SFS:emlxs0: /devices/pci@7c0/pci@0/pci@1/pci@0,2/SUNW,emlxs@1/fp@0,0 (CONNECTED)
2. FCT:emlxs1: /devices/pci@7c0/pci@0/pci@1/pci@0,2/SUNW,emlxs@1,1 (CONNECTED)

Enter an HBA number or zero to exit:
```

link_status <d_id>

Requests and returns the current link error status from the FC device specified by the D_ID address.

Example

```
emlxadm> link_status e8

D_ID: e8
            Link failures: 3 (0x3)
       Loss of sync count: 12 (0xc)
     Loss of signal count: 0 (0x0)
Primitive sequence errors: 0 (0x0)
         Invalid tx words: 17 (0x11)
        Invalid CRC count: 0 (0x0)
```

msgbuf [all], <number> [-i interval]

This command displays the current driver log, with various options.

Parameters

`[all]`	If the [all] parameter is specified, this command displays all of the current driver message log.
`<number>`	If the <number> parameter is specified, this command displays the last <number> of lines of the current driver message log.
`[-i interval]`	Using the [-i interval] argument enables the screen to be refreshed every [interval] of seconds. If the [-i interval] argument is not provided, the driver message log is displayed, followed by the "emlxadm>" prompt.

To stop the command from displaying the current driver log, press <Ctrl> and <C> at the same time.

Example

In this example, the last ten lines of the current driver log is displayed.

```
emlxadm> msgbuf 10

  155130.01: 1002033:[B.1C35]emlxs0:  DEBUG: 800: ELS sent.  (GA_NXT: did=fffffc
                                                             [00011000,00000000])
  155130.02: 1002034:[4.00C9]emlxs0:  DEBUG: 801: ELS comp.  (GA_NXT: CT_ACC: Rsn=0 Exp=0
                                                             [020113e1,21000020])
  155130.02: 1002035:[B.1C35]emlxs0:  DEBUG: 800: ELS sent.  (GA_NXT: did=fffffc
                                                             [000113e1,00000000])
  155130.02: 1002036:[4.00C9]emlxs0:  DEBUG: 801: ELS comp.  (GA_NXT: CT_ACC: Rsn=0 Exp=0
                                                             [020113e2,21000020])
  155130.02: 1002037:[B.1C35]emlxs0:  DEBUG: 800: ELS sent.  (GA_NXT: did=fffffc
                                                             [000113e2,00000000])
```

```
  155130.02: 1002038:[4.00C9]emlxs0:  DEBUG: 801: ELS comp.  (GA_NXT: CT_ACC: Rsn=0 Exp=0
                                                              [020113e4,21000020])
  155130.03: 1002039:[B.1C35]emlxs0:  DEBUG: 800: ELS sent.  (GA_NXT: did=fffffc
                                                              [000113e4,00000000])
  155130.03: 1002040:[4.00C9]emlxs0:  DEBUG: 801: ELS comp.  (GA_NXT: CT_ACC: Rsn=0 Exp=0
                                                              [020113e8,21000020])
  155130.03: 1002041:[B.1C35]emlxs0:  DEBUG: 800: ELS sent.  (GA_NXT: did=fffffc
                                                              [000113e8,00000000])
emlxadm>
```

ns

Performs and returns a complete query of the fabric name server.

Example

```
emlxadm> ns

Nameserver:
-------------------------------------------------------------------
     TYPE: Lport
      PID: 0113E1
     WWPN: 21000020371938fa
PORT_NAME: (SEAGATE ST39103FC       0004)
     WWNN: 20000020371938fa
NODE_NAME: (null)
      IPA: ffffffffffffffff
  IP_ADDR: 0.0.0.0
    CLASS: Class3
FC4_TYPES: 00000100,00000000,00000000,00000000,00000000,00000000,00000000,00000000
-------------------------------------------------------------------
     TYPE: Lport
      PID: 0113E2
     WWPN: 21000020371939a2
PORT_NAME: (SEAGATE ST39103FC       0004)
     WWNN: 20000020371939a2
NODE_NAME: (null)
IPA: ffffffffffffffff
  IP_ADDR: 0.0.0.0
    CLASS: Class3
FC4_TYPES: 00000100,00000000,00000000,00000000,00000000,00000000,00000000,00000000
-------------------------------------------------------------------
     TYPE: Lport
      PID: 0113E4
     WWPN: 21000020371938a3
PORT_NAME: (SEAGATE ST39103FC       0004)
     WWNN: 20000020371938a3
NODE_NAME: (null)
      IPA: ffffffffffffffff
  IP_ADDR: 0.0.0.0
    CLASS: Class3
FC4_TYPES: 00000100,00000000,00000000,00000000,00000000,00000000,00000000,00000000
```

```
------------------------------------------------------------------
     TYPE: Lport
      PID: 0113E8
     WWPN: 2100002037193670
PORT_NAME: (SEAGATE ST39103FC       0004)
     WWNN: 2000002037193670
NODE_NAME: (null)
      IPA: ffffffffffffffff
  IP_ADDR: 0.0.0.0
    CLASS: Class3
FC4_TYPES: 00000100,00000000,00000000,00000000,00000000,00000000,00000000,00000000
```

p

Repeats the last command.

Example

```
emlxadm> get_num_devs
There are 4 devices reported on this port.

emlxadm> p
emlxadm> get_num_devs

There are 4 devices reported on this port.
```

parm_get_num

Returns the total number of configurable parameters.

Example

```
emlxadm> parm_get_num
Result: There are 18 configurable parameters in the driver.
```

parm_get <label>

Gets the value of a specified parameter in the driver.

Example

```
emlxadm> parm_get adisc-support

  label: adisc-support
    min: 0
current: 1
    max: 2
default: 1
dynamic: yes
```

```
    desc: Sets the Fibre Channel ADISC login support level.
```

parm_get_list

Returns a list of configurable parameters.

Example

```
emlxadm> parm_get_list

Parameter:
------------------------------------------------------------------
  label: console-notices
    min: 0x0
current: 0x0
    max: 0xffffffff
default: 0x0
dynamic: yes
   desc: Verbose mask for notice messages to the console.
------------------------------------------------------------------
  label: console-warnings
    min: 0x0
current: 0x0
    max: 0xffffffff
default: 0x0
dynamic: yes
   desc: Verbose mask for warning messages to the console.
------------------------------------------------------------------
  label: console-errors
    min: 0x0
current: 0x0
    max: 0xffffffff
default: 0x0
dynamic: yes
   desc: Verbose mask for error messages to the console.
------------------------------------------------------------------
  label: log-notices
    min: 0x0
current: 0xffffffff
    max: 0xffffffff
default: 0xffffffff
dynamic: yes
   desc: Verbose mask for notice messages to the messages file.
------------------------------------------------------------------
  label: log-warnings
    min: 0x0
current: 0xffffffff
    max: 0xffffffff
default: 0xffffffff
```

```
dynamic: yes
   desc: Verbose mask for warning messages to the messages file.
----------------------------------------------------------------
  label: log-errors
    min: 0x0
current: 0xffffffff
    max: 0xffffffff
default: 0xffffffff
dynamic: yes
   desc: Verbose mask for error messages to the messages file.
----------------------------------------------------------------
  label: num-iocbs
    min: 128
current: 1024
    max: 10240
default: 1024
dynamic: no
   desc: Number of outstanding IOCBs driver can queue to adapter.
----------------------------------------------------------------
  label: ub-bufs
    min: 40
current: 1000
    max: 16320
default: 1000
dynamic: no
   desc: Number of unsolicited buffers the driver should allocate.
----------------------------------------------------------------
  label: network-on
    min: 0
current: 1
    max: 1
default: 1
dynamic: no
   desc: Enable IP processing.
----------------------------------------------------------------
  label: ack0
    min: 0
current: 0
    max: 1
default: 0
dynamic: no
   desc: Enable ACK0 support.
----------------------------------------------------------------
  label: topology
    min: 0
current: 0
    max: 6
default: 0
```

```
dynamic: no
   desc: Select Fibre Channel topology.
------------------------------------------------------------------
  label: link-speed
    min: 0
current: 0
    max: 4
default: 0
dynamic: no
   desc: Select link speed.
------------------------------------------------------------------
  label: num-nodes
    min: 2
current: 512
    max: 512
default: 512
dynamic: no
   desc: Number of fibre channel nodes (NPorts) the driver will support.
------------------------------------------------------------------
  label: cr-delay
    min: 0
current: 0
    max: 63
default: 0
dynamic: no
   desc: A count of milliseconds after which an interrupt response is generated.
------------------------------------------------------------------
  label: cr-count
    min: 1
current: 1
    max: 255
default: 1
dynamic: no
   desc: A count of I/O completions after which an interrupt response is
generated.
------------------------------------------------------------------
  label: assign-alpa
    min: 0x0
current: 0x0
    max: 0xef
default: 0x0
dynamic: no
   desc: Assigns a preferred ALPA to the port. Only used in Loop topology.
------------------------------------------------------------------
  label: adisc-support
    min: 0
current: 1
    max: 2
```

```
default: 1
dynamic: yes
   desc: Sets the Fibre Channel ADISC login support level.
------------------------------------------------------------------
  label: pm-support
    min: 0
current: 1
    max: 1
default: 1
dynamic: no
   desc: Enables power management support.
```

parm_set <label> <value>

Sets the value of a specified parameter in the driver. Only dynamic parameters can be set.

Note: To make this change permanent, you must edit the /kernel/drv/emlxs.conf file.

Examples

This example sets a dynamic parameter:

```
emlxadm> parm_set adisc-support 2

  label: adisc-support
    min: 0
current: 2
    max: 2
default: 1
dynamic: yes
   desc: Sets the Fibre Channel ADISC login support level.
```

This example attempts to set a static parameter (which is not allowed):

```
emlxadm> parm_set network-on 1
emlxadm: EMLX_PARM_SET: Parameter (network-on) is not dynamic and
cannot be changed here.

** To make this change you must edit the /kernel/drv/emlxs.conf or
**
** the /kernel/drv/emlx.conf file(s) and reboot the system.
```

q

Exits (quits) the utility program.

Example

```
emlxadm> q
```

```
Exiting...
```

reset_hard

Forces the adapter to perform a hardware reset.

Example

```
emlxadm> reset_hard
Done.
```

reset_link <[wwpn] or 0>

If the [wwpn] parameter is specified, this command resets the link of the specified FC device on the network. If the specified [wwpn] applies to a remote port, the reset link [wwpn] only works if the remote port is on a public loop. SFS uses the Loop Initialize ELS command to reset the link on a remote port. The LINIT ELS command is valid only if the remote N_Port is on a public loop.

If "0" (zero) is specified, this command resets the local link.

Examples

Resets the link of the specified FC device:

```
emlxadm> reset_link 21000020371938fa
Done.
```

Resets the local link:

```
emlxadm> reset_link 0
Done.
```

set_sym_nname <"string">

Sets the symbolic FC node name of the adapter to the string provided.

Note: This operation is not supported by the Solaris SFS stack.

Example

```
emlxadm> set_sym_nname "Emulex Corporation"
ioctl: FCIO_SET_SYM_NNAME: Operation not supported
```

set_sym_pname <"string">

Sets the symbolic FC port name of the adapter to the string provided.

Note: This operation is not supported by the Solaris SFS stack.

Example

```
emlxadm> set_sym_pname "Emulex Corporation"
ioctl: FCIO_SET_SYM_PNAME: Operation not supported
```

set_throttle

Temporarily sets the I/O queue <depth> for a specified remote port(s). The <depth> is the maximum number of concurrent I/Os that the driver can send to the specified remote port. The default queue depth for an FCP target port is specified by the "target-depth" driver parameter.

Command Options

`set_throttle all <depth>`	Sets the <depth> for all ports.
`set_throttle fcp <depth>`	Sets the <depth> for all FCP target ports.
`set_throttle wwpn <depth>`	Sets the <depth> for a specific port.

Example

This example sets the maximum number of concurrent I/Os to 512 for all remote ports.

```
emlxadm> set_throttle all 512

            ____________WWPN: Depth
            21000011c6810947: 512
            21000011c681065f: 512
            21000011c68108c0: 512
            21000011c681061d: 512
```

4. Using the emlxdrv Utility

The emlxdrv utility binds (associates) the following FC and NIC drivers to various Emulex FC and NIC adapter models, respectively:

FC Drivers

- emlxs - Solaris inbox FC/FCoE driver
- elxfc - Emulex-distributed FC/FCoE Solaris driver (does not support Oracle-branded devices)
- lpfc - Legacy Emulex distributed FC SD driver, which does not support Oracle-branded devices. The lpfc driver is available for Solaris 10, but not for Solaris 11.

NIC Drivers

- elxnic - Emulex-distributed Solaris NIC driver
- oce - Solaris inbox NIC driver

The emlxs and elxfc FC drivers can coexist on the same host. However, a specific FC adapter model is associated to only one of the FC drivers (emlxs or elxfc) at a time. Likewise, a specific NIC adapter model is associated to only one of the NIC drivers (elxnic or oce) at a time.

Notes:

- The simultaneous use of the lpfc driver and the emlxs or the elxfc driver is not supported. Using the lpfc driver should only be used as a step when migrating complex configurations between drivers.
- If you change the driver binding configuration, the emlxdrv utility requires a system reboot in order for the new configuration to take effect.
- Oracle-branded 16Gb adapters and the standard models share the same device id, therefore it is not possible to bind them to two different drivers. Both adapters must be bound to the same driver.
- If an Oracle-branded 16Gb universal host bus adapter is detected on the system, the emlxdrv utility will only allow the adapter to bind with the emlxs driver.

The emlxdrv utility program can run in two modes:

- Interactive
- CLI

Interactive Mode of Operation for emlxdrv

To run the emlxadm utility in interactive mode, type “emlxdrv” without any commands or arguments. For example:

```
# emlxdrv
```

Displaying Bindings Between Drivers and Adapter Models

The emlxdrv program scans the host system and prepares a driver configuration table consisting of bindings (associations) between the FC and NIC drivers (emlxs, elxfc, lpfc, oce, and elxnic) and a list of Emulex FC and NIC adapter models. After the table is prepared, the utility displays:

- FC driver configuration table
- NIC driver configuration table
- List of available commands
- emlxdrv prompt

For example, after "emlxdrv" is entered, the utility displays the following:

```
FC Driver Alias Pres Boot Oracle emlxs  elxfc lpfc  Models
elxfc     lpfs  -    -    -      yes    yes   yes   LP8000S, LP9002S (SBUS)
elxfc     f800  -    -    -      yes    yes   yes   LP8000, LP8000DC
elxfc     f900  -    -    -      yes    yes   yes   LP9002, LP9002C,
                                                    LP9002DC, LP9402DC
elxfc     f980  -    -    -      yes    yes   yes   LP9802,LP9802DC
elxfc     fa00  -    -    -      yes    yes   yes   LP10000, LP10000DC,
                                                    LP10000ExDC
elxfc     fd00  -    -    -      yes    yes   yes   LP11000, LP11002
elxfc     fe00  yes  -    -      yes    yes   yes   LPe11000, LPe11002,
                                                    LPe11004
elxfc     fe12  -    -    -      yes    yes   yes   LPe11000, LPe11002,
                                                    LPe11004
elxfc     f100  -    -    -      yes    yes   yes   LPe12000, LPe12002
elxfc     f111  -    -    -      yes    yes   yes   LPe12000, LPe12002
elxfc     f112  -    -    -      yes    yes   yes   LPe12000, LPe12002
elxfc     fe05  -    -    -      yes    yes   yes   LP21000, LP21002
elxfc     f0a5  -    -    -      yes    yes   -     2G Blade Adapter
emlxs     fc00  -    -    yes    yes    -     -     LP10000-S, LP10000DC-S
emlxs     fc10  yes  -    yes    yes    -     -     LP11000-S, LP11002-S
emlxs     fc20  -    -    yes    yes    -     -     LPe11000-S, LPe11002-S
emxls     fc40  -    -    yes    yes    -     -     LPe12000-S, LPe12002-S
elxfc     704   -    -    -      yes    yes   -     OCe10101, OCe10102
elxfc     714   yes  -    -      yes    yes   -     OCe11101, OCe11102


NIC Driver Alias   Pres  Oracle  oce    elxnic Models
oce        700     -     -       yes    yes    OCe10101, OCe10102
elxnic     710     -     -       yes    yes    OCe11101, OCe11102
```

```
Available Commands
set...     - Sets driver bindings to specified devices.
clear...   - Clears driver bindings to specified devices.
refresh...- Refresh driver bindings to current devices.
q          - Exits this program.
```

emlxdrv>

The definitions of headings for the FC driver configuration table:

Table 4-1 FC Driver Configuration Table Heading Descriptions

Heading	Description
FC Driver	The type of FC driver (emlxs, elxfc, lpfc, or "-" for none) that is currently configured to bind or attach to a specific adapter alias.
Alias	The specific adapter alias associated with a set of Emulex adapter models. Driver bindings can be made only with a specific adapter alias and not with a specific adapter model.
Pres	A "yes" indicates that this type of adapter is currently present in the host system. The emlxdrv utility allows you to bind a driver to adapters that are not currently present in the system, but may be present in the future.
Boot	A "yes" indicates that this specific type of adapter is currently providing connectivity to the system's boot disk. **Note:** The emlxdrv utility does not allow you to change the driver binding to an adapter currently providing connectivity to the boot disk. If the driver binding needs to be changed to a boot device, the system must first be configured to boot through an adapter of another type. This procedure is not in the scope of this document.
Oracle	A "yes" indicates that this specific type of adapter is branded and sold directly by Oracle.
emlxs	A "yes" indicates the emlxs driver supports this specific type of adapter.
elxfc	A "yes" indicates the elxfc driver supports this specific type of adapter.
lpfc	A "yes" indicates the lpfc driver supports this specific type of adapter.
Models	A list of Emulex adapter models that are identified by a common adapter alias. Driver bindings can be made only with a specific adapter alias and not with a specific adapter model.

The definitions of headings for the NIC driver configuration table:

Table 4-2 NIC Driver Configuration Table Heading Descriptions

Heading	Description
NIC Driver	The type of NIC driver (oce or elxnic) that is currently configured to bind or attach to a specific adapter alias.
Alias	See the "Alias" description in Table 4-1 on page 54.
Pres	See the "Pres" description in Table 4-1 on page 54.
Oracle	See the "Oracle" description in Table 4-1 on page 54.

Table 4-2 NIC Driver Configuration Table Heading Descriptions (Continued)

Heading	Description
oce	A “yes” indicates the oce driver supports this specific type of adapter.
elxnic	A “yes” indicates the elxnic driver supports this specific type of adapter.
Models	See the “Models” description in Table 4-1 on page 54.

Entering emlxdrv Commands

After the available commands are listed, the “emlxdrv>” prompt is displayed. From this point, the utility is prompt-driven. When the prompt is displayed, you can enter one of the commands in the list. For example, to refresh the host bindings and main display with the latest information, use the “refresh” command:

```
emlxdrv> refresh
```

The current driver configuration table and the available command list are displayed automatically after each command is issued.

Some commands require a <driver> or <driver_name> and an <alias> argument. You must specify one of the valid adapter aliases listed in the current driver configuration table. Each alias is shared by multiple adapter models. Driver bindings can be made only with an adapter alias and not with a specific adapter model.

Exiting emlxdrv

To exit (quit) the emlxdrv utility, type “q” at the prompt:

```
emlxdrv> q
```

CLI Mode of Operation for emlxdrv

You can run the emlxdrv utility program in CLI mode by typing the name of the program followed by a valid command and any required command arguments. For example:

```
# emlxdrv refresh
```

This mode of operation enables you to use the emlxdrv utility as part of a script or another program capable of running system-level calls.

For example, you can revise the device binding by entering all the information on one line at the operating system prompt as shown below.

```
FC Driver Alias Pres Boot Oracle emlxs  elxfc lpfc  Models
elxfc     lpfs  -    -    -      yes    yes   yes   LP8000S, LP9002S (SBUS)
elxfc     f800  -    -    -      yes    yes   yes   LP8000, LP8000DC
elxfc     f900  -    -    -      yes    yes   yes   LP9002, LP9002C,
                                                    LP9002DC, LP9402DC
elxfc     f980  -    -    -      yes    yes   yes   LP9802,LP9802DC
```

```
elxfc     fa00  -    -    -      yes    yes    yes    LP10000, LP10000DC,
                                                      LP10000ExDC
elxfc     fd00  -    -    -      yes    yes    yes    LP11000, LP11002
elxfc     fe00  yes  -    -      yes    yes    yes    LPe11000, LPe11002,
                                                      LPe11004
elxfc     fe12  -    -    -      yes    yes    yes    LPe11000, LPe11002,
                                                      LPe11004
elxfc     f100  -    -    -      yes    yes    yes    LPe12000, LPe12002
elxfc     f111  -    -    -      yes    yes    yes    LPe12000, LPe12002
elxfc     f112  -    -    -      yes    yes    yes    LPe12000, LPe12002
elxfc     fe05  -    -    -      yes    yes    yes    LP21000, LP21002
elxfc     f0a5  -    -    -      yes    yes    -      2G Blade Adapter
emlxs     fc00  -    -    yes    yes    -      -      LP10000-S, LP10000DC-S
emlxs     fc10  yes  -    yes    yes    -      -      LP11000-S, LP11002-S
emlxs     fc20  -    -    yes    yes    -      -      LPe11000-S, LPe11002-S
emxls     fc40  -    -    yes    yes    -      -      LPe12000-S, LPe12002-S
elxfc     704   -    -    -      yes    yes    -      OCe10101, OCe10102
elxfc     714   yes  -    -      yes    yes    -      OCe11101, OCe11102

NIC Driver Alias    Pres  Oracle  oce    elxnic Models
oce        700      -     -       yes    yes    OCe10101, OCe10102
elxnic     710      -     -       yes    yes    OCe11101, OCe11102
```

Command Descriptions for emlxdrv

Note: You can view the list of commands at any time by running the emlxdrv utility in interactive mode (see "Interactive Mode of Operation for emlxdrv" on page 52).

This section provides a list of commands that can be issued with the emlxdrv utility program. Table 4-3 summarizes this command list, including abbreviated descriptions.

Table 4-3 Summary of emlxdrv Commands

Command	Description	Page
`clear...`	Clears the bindings from the specified driver(s) to the specified device(s). The clear command has several options:	57
	• clear [driver] all – clears bindings from the specified [driver] to all devices.	57
	• clear dev [alias] - clears bindings from a valid adapter alias to its associated devices.	59
	• clear [driver] emulex -clears bindings from the specified [driver] to all Emulex-branded devices.	59
	• clear [driver] oracle – clears bindings from the specified [driver] to all Oracle-branded devices. The valid [driver] values are fc, nic, and [driver_name].	60
`q`	Exits (quits) the emlxdrv utility.	60
`refresh`	Refreshes the host bindings and main display with the latest information.	61
`set...`	Sets the specific driver (emlxs, elxfc, lpfc, elxnic, or oce) to bind to the specified device(s). The set command has several options:	62
	• set [driver_name] all – sets the [driver_name] to bind to all devices.	62
	• set [driver_name] dev [alias] - sets the [driver_name] to bind to the devices specified by a valid adapter alias.	62
	• set [driver_name] emulex - sets the [driver_name] to bind to all Emulex-branded devices.	63
	• set [driver_name] oracle – sets the [driver_name] to bind to all Oracle-branded devices. The valid [driver_name] values are emlxs, elxfc, lpfc, elxnic, or oce.	63

clear...

Clears the bindings from the specified driver(s) to the specified device(s). This command has various options as defined in the following subsections.

Note: When running a clear command, you may see a "Cannot unload module..." message. This message indicates that you must reboot the system to unbind a driver from that adapter alias. The emlxdrv utility requires a system reboot for the new configuration take effect.

clear [driver] all

Clears the bindings from the specified [driver] to all devices. The valid [driver] values are fc, nic, and [driver_name], as defined in the following section.

Command Options

`clear all`	Clears bindings from all drivers to all devices. This is the default value for the clear command.
`clear fc all`	Clears bindings from FC drivers to all devices.
`clear nic all`	Clears bindings from NIC drivers to all devices.
`clear [driver_name] all`	Clears bindings from a specific driver (indicated by its name) to all devices.

Example

```
emlxadm> clear all

Updating lpfs ...
Cannot unload module: emlxs
Will be unloaded upon reboot.

Updating f800 ...
Cannot unload module: emlxs
Will be unloaded upon reboot.

Updating f900 ...
Cannot unload module: lpfc
Will be unloaded upon reboot.

Updating f980 ...
Cannot unload module: lpfc
Will be unloaded upon reboot.

Updating fa00 ...
Cannot unload module: emlxs
Will be unloaded upon reboot.

Updating fd00 ...
Cannot unload module: emlxs
Will be unloaded upon reboot.

Updating fe00 ...
Cannot unload module: emlxs
Will be unloaded upon reboot.

Done.
```

clear dev [alias]

Clears the bindings from a valid adapter alias, as listed in one of the driver configuration tables (see the "Alias" column in Table 4-1 on page 54), to its associated devices. Each alias is shared by multiple adapter models.

Example

```
emlxdrv> clear dev fe00

Updating fe00 ...
Cannot unload module: emlxs
Will be unloaded upon reboot.
Done.
```

clear [driver] emulex

Clears the bindings from the specified [driver] to all Emulex-branded devices. The valid [driver] values are fc, nic, and [driver_name], as defined in the following section.

Command Options

`clear emulex`	Clears bindings from all drivers to all Emulex-branded devices
`clear fc emulex`	Clears bindings from FC drivers to all Emulex-branded devices.
`clear nic emulex`	Clears bindings from all NIC drivers to all Emulex-branded devices.
`clear [driver_name] emulex`	Clears bindings from a specific driver (indicated by its name) to all Emulex-branded devices.

Example

```
emlxdrv> clear emulex

Cannot unload module: emlxs
Will be unloaded upon reboot.

Updating fc00 ...
Cannot unload module: emlxs
Will be unloaded upon reboot.

Updating fc10 ...
Cannot unload module: emlxs
Will be unloaded upon reboot.

Updating fc20 ...
Cannot unload module: emlxs
Will be unloaded upon reboot.
```

```
Done.
```

clear [driver] oracle

Clears bindings from the specified [driver] to all Oracle-branded devices. The valid [driver] values are fc, nic, and [driver_name], as defined in the following section.

Command Options

`clear oracle`	Clears bindings to all Oracle-branded devices for all drivers.
`clear fc oracle`	Clears bindings to all Oracle-branded devices for FC drivers.
`clear nic oracle`	Clears bindings to all Oracle-branded devices for NIC drivers
`clear [driver_name] oracle`	Clears bindings to all Oracle-branded devices for a specific driver.

Example

```
emlxdrv> clear oracle

Updating fc00 ...
Cannot unload module: emlxs
Will be unloaded upon reboot.

Updating fc10 ...
Cannot unload module: emlxs
Will be unloaded upon reboot.

Updating fc20 ...
Cannot unload module: emlxs
Will be unloaded upon reboot.
Done.
```

q

Exits (quits) the emlxdrv utility.

Note: If changes were made to the driver bindings, a system reboot is required for the changes to take effect. If NIC changes were made, a system reboot is also required.

Example

```
emlxdrv> q

Exiting...

NOTE: If changes were made, then a system reboot may be required.
```

refresh

Refreshes the host bindings and main display with the latest information.

Example

```
emlxdrv> refresh
```

The following output is displayed:

```
FC Driver   Alias Pres Boot Oracle emlxs  elxfc lpfc  Models
elxfc       lpfs  -    -    -      yes    yes   yes   LP8000S, LP9002S
                                                      (SBUS)
elxfc       f800  -    -    -      yes    yes   yes   LP8000, LP8000DC
elxfc       f900  -    -    -      yes    yes   yes   LP9002, LP9002C,
                                                      LP9002DC, LP9402DC
elxfc       f980  -    -    -      yes    yes   yes   LP9802,LP9802DC
elxfc       fa00  -    -    -      yes    yes   yes   LP10000, LP10000DC,
                                                      LP10000ExDC
elxfc       fd00  -    -    -      yes    yes   yes   LP11000, LP11002
elxfc       fe00  yes  -    -      yes    yes   yes   LPe11000, LPe11002,
                                                      LPe11004
elxfc       fe12  -    -    -      yes    yes   yes   LPe11000, LPe11002,
                                                      LPe11004
elxfc       f100  -    -    -      yes    yes   yes   LPe12000, LPe12002
elxfc       f111  -    -    -      yes    yes   yes   LPe12000, LPe12002
elxfc       f112  -    -    -      yes    yes   yes   LPe12000, LPe12002
elxfc       fe05  -    -    -      yes    yes   yes   LP21000, LP21002
elxfc       f0a5  -    -    -      yes    yes   -     2G Blade Adapter
emlxs       fc00  -    -    yes    yes    -     -     LP10000-S, LP10000DC-S
emlxs       fc10  yes  -    yes    yes    -     -     LP11000-S, LP11002-S
emlxs       fc20  -    -    yes    yes    -     -     LPe11000-S, LPe11002-S
emxls       fc40  -    -    yes    yes    -     -     LPe12000-S, LPe12002-S
elxfc       704   -    -    -      yes    yes   -     OCe10101, OCe10102
elxfc       714   yes  -    -      yes    yes   -     OCe11101, OCe11102


NIC Driver  Alias    Pres   Oracle   oce     elxnic Models
oce         700      -      -        yes     yes    OCe10101, OCe10102
elxnic      710      -      -        yes     yes    OCe11101, OCe11102


Available Commands
set...      - Sets driver bindings to specified devices.
clear...    - Clears driver bindings to specified devices.
refresh...  - Refresh driver bindings to current devices.
q           - Exits this program.
```

set...

This command sets the bindings from the specified driver (emlxs, elxfc, lpfc, elxnic, or oce) to the specified device(s). This command has various options as defined in the following subsections.

Note: When running a set command, you may see a "Cannot unload module..." message. This message indicates that you must reboot the system to unbind a driver from that adapter alias. The emlxdrv utility requires a system reboot for the new configuration take effect.

set [driver_name] all

Sets the specified [driver_name] to bind to all devices. The valid [driver_name] values are emlxs, elxfc, lpfc, elxnic, or oce.

Example

```
emlxdrv> set emlxs all

Updating lpfs ...
Updating f800 ...
Updating f900 ...
Cannot unload module: lpfc
Will be unloaded upon reboot.

Updating f980 ...
Cannot unload module: lpfc
Will be unloaded upon reboot.

Updating fa00 ...
Updating fd00 ...
Updating fe00 ...
Updating fc00 ...
Updating fc10 ...
Updating fc20 ...
Done.
```

set [driver_name] [alias]

Sets the specified [driver_name] to bind to the devices specified by a valid adapter alias (as listed in one of the driver configuration tables (see the "Alias" column in Table 4-1 on page 54). Note that each alias is shared by multiple adapter models. Driver bindings

can be made only with an adapter alias and not with a specific adapter model. The valid [driver_name] values are emlxs, elxfc, lpfc, elxnic, or oce.

Example

```
emlxdrv> set emlxs f980

Updating f980 ...
Cannot unload module: lpfc
Will be unloaded upon reboot.

Done.
```

set [driver_name] emulex

Sets the specific [driver_name] to bind to all Emulex-branded devices. The valid [driver_name] values are emlxs, elxfc, lpfc, elxnic, or oce.

Example

```
emlxdrv> set lpfc fa00

Updating fa00 ...
Cannot unload module: emlxs
Will be unloaded upon reboot.
Done.
```

set [driver_name] oracle

Sets the specific [driver_name] to bind to all Oracle-branded devices. The valid [driver_name] values are emlxs, elxfc, lpfc, elxnic, or oce.

Example

```
emlxdrv> set emlxs oracle

Updating fc00 ...
Updating fc10 ...
Updating fc20 ...
Done.
```