



EMULEX[®]

We network storage

FCA Utilities

Reference Manual

Version 1.00m

for use with the

*Emulex[®]-Sun LightPulse[®] Fibre Channel Adapter Driver
(emlxs)*

Copyright© 2005 Emulex Corporation. All rights reserved worldwide. No part of this document may be reproduced by any means nor translated to any electronic medium without the written consent of Emulex Corporation.

Information furnished by Emulex Corporation is believed to be accurate and reliable. However, no responsibility is assumed by Emulex Corporation for its use; or for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Emulex Corporation.

Emulex and LightPulse are registered trademarks, and AutoPilot Installer, AutoPilot Manager, BlockGuard, FibreSpy, HBAnyware, InSpeed, MultiPulse, SLI and SBOD are trademarks, of Emulex Corporation. All other brand or product names referenced herein are trademarks or registered trademarks of their respective companies or organizations.

Emulex provides this manual "as is" without any warranty of any kind, either expressed or implied, including but not limited to the implied warranties of merchantability or fitness for a particular purpose. Emulex Corporation may make improvements and changes to the product described in this manual at any time and without any notice. Emulex Corporation assumes no responsibility for its use, nor for any infringements of patents or other rights of third parties that may result. Periodic changes are made to information contained herein; although these changes will be incorporated into new editions of this manual, Emulex Corporation disclaims any undertaking to give notice of such changes.

Introduction	1
Fibre Channel Overview	1
The Solaris Fibre Channel Stack	1
Installing the Utilities	3
Installing or Updating the Utilities Using the emlXu_install Script	3
Removing the Utilities Using the emlXu_remove Script.....	4
Installing the Utilities Package Manually	5
Removing the Utilities Package Manually	5
Updating the Utilities Package Manually.....	6
Using the emlXadm Utility	7
Modes of Operation (emlXadm)	7
Interactive Mode (emlXadm)	7
CLI Mode (emlXadm).....	8
Command Descriptions (emlXadm).....	8
get_num_devs.....	9
get_dev_list.....	9
get_logi_params <wwpn>.....	9
get_host_params.....	11
get_sym_pname.....	11
set_sym_pname <"string">.....	11
get_sym_nname.....	11
set_sym_nname <"string">.....	11
dev_login <wwpn>.....	12
dev_logout.....	12
get_state <wwpn>	12
dev_remove <wwpn>	12
link_status <d_id>	12
get_fcode_rev	12
download_fcode <filename>.....	13
get_fw_rev.....	13
download_fw <filename>	13
get_boot_rev	13
download_boot <filename>	14
get_dump_size	14
force_dump	14
get_dump <-t filename.txt or -b filename.bin>	14
get_topology.....	15
reset_link <wwpn or zero for local link>.....	15
reset_hard.....	15
reset_hard_core	15
diag <test [parameters]> or diag code <cmd_code (hex)>	15
ns	16
parm_get_num	17
parm_get_list.....	17
parm_get <label>	19
parm_set <label> <value>	20
msgbuf all or <number> [-i interval]	20
get_host_attrs	21
get_port_attrs <index>, <wwn> or all.....	21
get_path <index>.....	24
q.....	24

h	24
hba	25
p	25
Using the emlxdrv Utility	26
Modes of Operation (emlxdrv)	26
Interactive Mode (emlxdrv)	26
CLI Mode (emlxdrv)	27
Command Descriptions (emlxdrv)	28
set_emlxs <alias>	28
set_emlxs_sun	28
set_emlxs_all	28
set_lpfcs <alias>	29
set_lpfcs_nonsun	29
clear_dev <alias>	29
clear_lpfcs	30
clear_emlxs	30
clear_sun	30
clear_nonsun	30
clear_all	31
q	32

Introduction

This document provides the information needed to use the Emulex® emlxadm and elmxdrv utility programs. For system administrators, this document includes information about the installation and removal of these utilities.

At the very least, system administrators should be familiar with Solaris and the Sun StorEdge SAN Foundation Software (SFS) and should have access to standard system documentation. Anyone working with this product should have some familiarity with the nature and use of Fibre Channel.

Fibre Channel Overview

Fibre Channel is a general-purpose, high-throughput, low-latency interconnect. It employs serial bit transmission over copper media, short-wave and long-wave optical media. Cable distances may range up to 30 meters for copper media and up to 10 kilometers for optical media. Transmission speeds currently range up to 4 gigabits per second, or roughly 400 megabytes per second. Fibre Channel transmission protocols provide high reliability, with bit error rates less than 1 in 10^{12} . Interconnects may be configured as point-to-point, loop, or fabric (network switch).

The Fibre Channel specifications provide for the emulation of two traditional protocols: SCSI and IP. For storage, Fibre Channel provides an emulation of SCSI; this emulation is dubbed FCP, short for *Fibre Channel Protocol* – a confusing acronym, in that it refers specifically to SCSI-on-Fibre Channel rather than to the lower-level protocols by which Fibre Channel itself operates. Throughout this document, we generally refer to FCP; when we mention SCSI, we are referring to the particular SCSI properties within FCP, or to the original SCSI protocol. For networking, Fibre Channel provides an encapsulation of IP (Internet Protocol), referred to in this document as *IP*.

The Solaris Fibre Channel Stack

Each Fibre Channel host bus adapter (HBA) is managed by an associated device driver. A device driver acts as a translator between an operating system and the hardware so that the operating system's kernel need not know the specifics of the device it uses. A device driver contains all of the code specific to operating a device and provides an input/output (I/O) interface to the rest of the system.

The Emulex-Sun LightPulse® HBA device driver for Solaris, emlxs, is a Fibre Channel adapter (FCA) driver as specified by the Sun Fibre Channel architecture (also known as Leadville). The heart of the Sun Fibre Channel architecture consists of the Fibre Channel transport layer (FCTL) modules, which provide a common interface for various Fibre Channel adapters on a host. These modules consist of several tightly coupled pieces, including a per-port driver (FP) and a system-wide transport layer (FCTL) driver. The FP driver handles all per-port state and common services needed by a variety of protocols and the FCA drivers. The FCTL module provides consistent, system-wide access of Fibre Channel devices and services to upper layer protocols (ULP) and administration utilities. The point of providing a Fibre Channel port/transport interface is to abstract and define all services available through an FCA driver needed by FC-4 ULP drivers such as the SCSI driver (FCP) and the IP driver (FCIP). The FCP function provides access to Fibre Channel disk and tape drives. The IP function provides peer-to-peer networking, such as TCP or UDP, between Fibre Channel hosts.

Figure 1 shows the software stack for network operations and SCSI operations from the standpoint of a Solaris host.

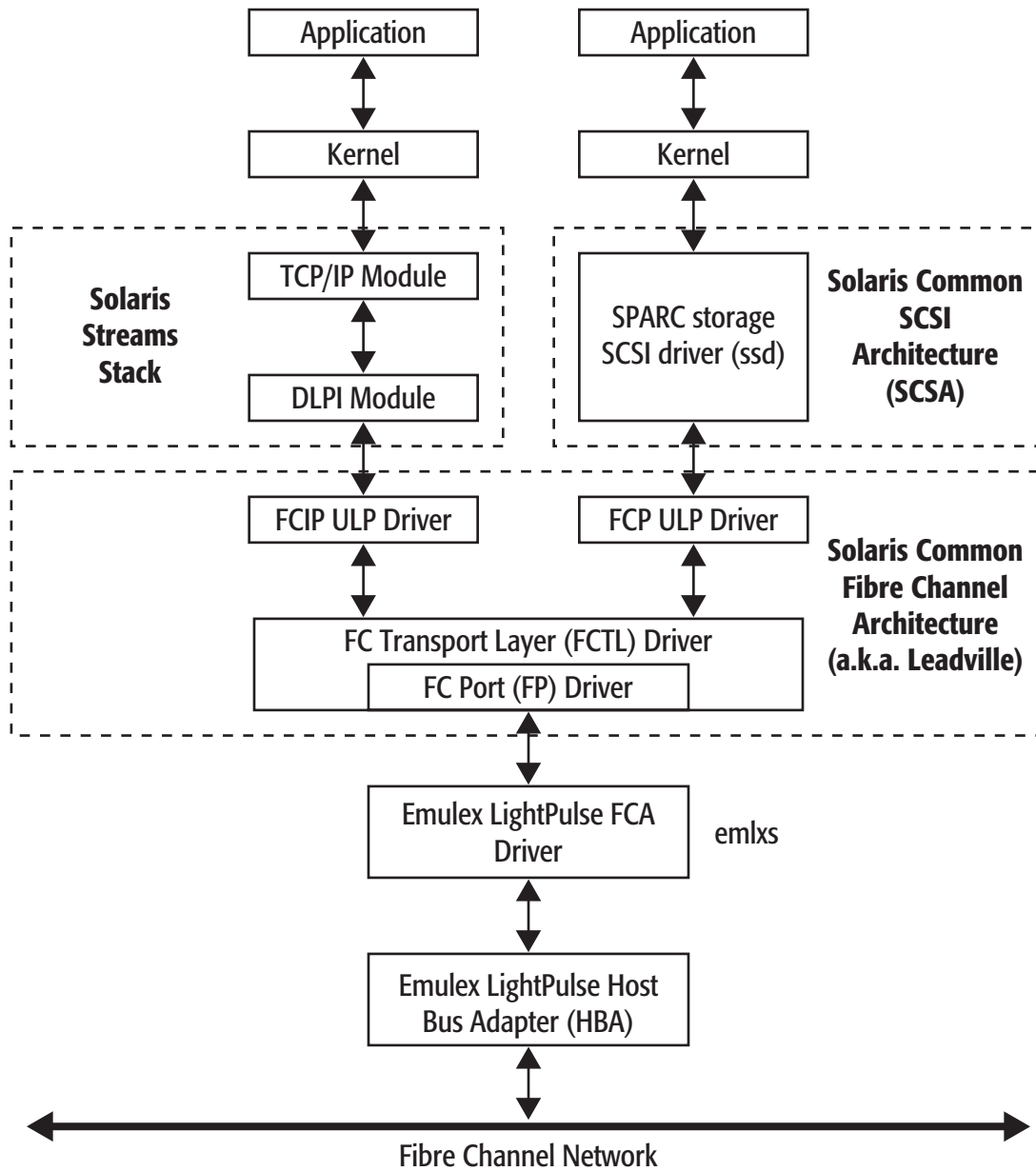


Figure 1: The Solaris Software Stack

Installing the Utilities

Caution: Before installing the Emulex utilities package, you must first install the Sun StorEdge SAN Foundation Software package and all the recommended patches as described in the *Sun StorEdge SAN Foundation Software Installation Guide* provided by Sun.

The Emulex utility programs are provided as part of the emlxu utilities kit. You can download the emlxu utilities kit (for example, emlxu_kit-1.00x-sparc.tar for SPARC platforms or emlxu_kit-1.00x-i386.tar for x86/x64 platforms) from the Sun page of the Emulex Web site (<http://www.emulex.com/ts/docoem/sun/10k.htm>).

Installing or Updating the Utilities Using the emlxu_install Script

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

Before installing the Emulex emlxu utilities package, you must have completely installed the Sun StorEdge SAN Foundation Software package, all the recommended patches as described in the *Sun StorEdge SAN Foundation Software Installation Guide*, and the Emulex-Sun Fibre Channel adapter driver package SUNWemlxs.

If an earlier version of the emlxu utilities package is already installed on the system, the emlxu_install script will first remove the old version before installing the new version.

To install the utilities kit using the emlxu_install script, do the following:

1. Log in as root, or su to root.
2. Copy the utilities kit from your distribution medium into a directory, referred to here as <directory>. The utilities kit is a .tar file named something similar to emlxu_kit-1.00x-sparc.tar.
3. Change to the directory where you put the kit tar file by typing

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

```
tar xf emlxu_kit-1.00x-sparc.tar emlxu_install
```
5. Install the kit by typing

```
emlxu_install
```
6. The script removes any earlier version of the emlxu utilities package. (If an earlier package is not found, this fact is indicated; skip to step 9.) The following text is displayed:

```
<Removing old EMLXemlxu package>
```
7. If an old package is installed, you are prompted to remove it:

```
Do you want to remove this package? [y,n,?,q]
```
8. Enter **y**. The following message is displayed:

```
Removal of <EMLXemlxu> was successful.
```
9. The script expands the utilities kit .tar file and begins installing the new package. The following message is displayed:

```
<Expanding emlxu_kit-1.00x-sparc.tar>
<Adding new package>
```
10. The script installs the emlxu utilities package. The package is prepared for installation and you are prompted for confirmation by the following message:

```
Do you want to continue with the installation of <EMLXemlxu> [y,n,?]:
```
11. Enter **y**. The installation package provides running commentary on the installation process.

12. 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.
```

13. The script performs some cleanup and displays the following messages:

```
<Cleaning directory>  
<emlXu install complete>  
<Execute "emlXu_remove" when ready to uninstall>
```

14. The script leaves a copy of the emlXu_remove script in your working directory with the original utilities kit tar file. You can remove this script, or leave it in the directory if you may want to uninstall the emlXu utilities from your system in the future. See *Removing the Utilities Using the emlXu_remove Script* on page 4 for more details.

The emlXu utilities installation is complete. The utility package's programs are located in the /opt/EMLXemlXu/bin directory.

You do not have to reboot your system to begin running a utility program. However, to run a program you must either enter the program's full path name, or add the package's bin directory (/opt/EMLXemlXu/bin) to your 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 your environment's man path.

For further information on installing and removing packages, consult the Solaris system administration documentation and the pkgadd(1M) and pkgrm(1M) manual pages.

Removing the Utilities Using the emlXu_remove Script

You can uninstall the utilities kit using the emlXu_remove script. If you do not have the emlXu_remove script and you do not have the original emlXu utilities kit tar file, you must uninstall the emlXu package manually; follow the instructions in *Removing the Utilities Package Manually* on page 5. If you are updating the emlXu utilities to a newer version and you have the new utilities kit tar file, you do not have to use the emlXu_remove script; the emlXu_install script removes any old version as it installs the newer version; see *Installing or Updating the Utilities Using the emlXu_install Script* on page 3 for more details.

If you do not want to update the utilities package, and only want to uninstall it, use the emlXu_remove script by doing the following (all emlXu files are removed):

1. Log in as root, or su to root.
2. Go to the directory where the emlXu_remove script is located, or to the directory where the original utilities kit tar file is located, by typing

```
cd <directory>
```
3. If you have the emlXu_remove script, skip to step 4. If you do not have the emlXu_remove script but you do have the original emlXu utilities kit tar file, extract the emlXu_remove script from the tar file by typing

```
tar xf emlXu_kit-1.00x-sparc.tar emlXu_remove
```
4. Remove the emlXu utilities package by typing

```
emlXu_remove
```
5. The script locates the EMLXemlXu utilities package, and the following message is displayed:

```
<Removing EMLXemlXu package>
```

If no package is installed, a message indicates this; skip to step 7. Otherwise, you are prompted to remove the package with the following message:

```
Do you want to remove this package? [y,n,?,q]
```
6. Enter **y**. The following message is displayed:

```
Removal of <EMLXemlXu> was successful.
```
7. The script performs some cleanup and displays the following message:


```
<Removing emlXu scripts>  
<emlXu_remove complete>
```

The utilities package has been removed. If you want to install another version of the emlXu utilities package, do so now by following the instructions in one of the following sections:

- *Installing or Updating the Utilities Using the emlXu_install Script* on page 3
- *Installing the Utilities Package Manually* on page 5

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

Installing the Utilities Package Manually

If an earlier version of the emlXu utilities package is already installed on the system and you want to install a different version, follow the instructions in *Removing the Utilities Package Manually* on page 5, then return to this section to install the new utilities package.

Caution: Before installing the Emulex utilities package, you must first install the Sun StorEdge SAN Foundation Software package and all the recommended patches as described in the *Sun StorEdge SAN Foundation Software Installation Guide* provided by Sun.

To install the emlXu utilities package manually, do the following:

1. Log in as root, or su to root.
2. Copy the utilities kit from your distribution medium into a directory, referred to here as <directory>. The utilities kit is a .tar file named something similar to emlXu_kit-1.00x-sparc.tar.
3. Change to the directory where you put the kit tar file by typing

```
cd <directory>
```
4. Extract the installation images from the tar file by typing

```
tar xvf emlXu_kit-1.00x-sparc.tar
```
5. install the EMLXemlXu utilities package by typing

```
pkgadd -d . EMLXemlXu
```
6. The package is prepared for installation, and you are prompted to confirm the installation with the following message:

```
Do you want to continue with the installation of <EMLXemlXu> [y,n,?]
```
7. Enter **y**. The installation package provides running commentary on the installation process.
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 emlXu utilities installation is complete. The utility package's programs are located in the /opt/EMLXemlXu/bin directory.

You do not have to reboot your system to begin running a utility program. However, to run a program you must either enter the program's full path name, or add the package's bin directory (/opt/EMLXemlXu/bin) to your 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 your environment's man path.

Removing the Utilities Package Manually

To remove the emlXu utilities package, do the following:

1. Remove the EMLXemlXu utilities package by typing

```
pkgrm EMLXemlXu
```

2. You are prompted to confirm the removal by the following message:

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

3. Enter **y**. The package is prepared for removal, and you are prompted again for confirmation:

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

4. Enter **y**. The following message is displayed:

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

The utilities package has been removed.

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

Updating the Utilities Package Manually

Update the `emlxu` utilities by doing the following:

1. Remove the old `emlxu` utilities package by following the instructions in *Removing the Utilities Package Manually* on page 5.
2. Install the new `emlxu` utilities package by following the instructions in *Installing the Utilities Package Manually* on page 5.

Using the emlxadm Utility

The emlxadm utility is intended to be a direct user interface to the Fibre Channel input/output (FCIO) interface provided by the Sun StorEdge SFS. The FCIO interface provides a Sun common ioctl interface to the FCTL, which manages the FCA drivers for each Fibre Channel HBA attached to the host system.

Modes of Operation (emlxadm)

The emlxadm utility program can be run in two modes:

- Interactive
- Command line interface (CLI)

Interactive Mode (emlxadm)

The emlxadm utility program can be run in an interactive command mode by typing the name of the program without any command line arguments:

```
# emlxadm
```

After it is started, the emlxadm program scans the host system and prepares a list of qualified HBA ports to choose from. Qualified HBA ports are devices that attach to the SUN StorEdge SFS through the FP driver. After the list is prepared, the utility displays the following:

```
EMLXADM Device Management Utility, Version 1.00q  
COPYRIGHT © 2004-2005 Emulex. All rights reserved.
```

```
Available HBA's:
```

1. /devices/pci@1e,600000/SUNW,qlc@3/fp@0,0:devctl (CONNECTED)
2. /devices/pci@1e,600000/SUNW,qlc@3,1/fp@0,0:devctl (NOT CONNECTED)
3. /devices/pci@1e,600000/SUNW,emlxs@2/fp@0,0:devctl (CONNECTED)
4. /devices/pci@1e,600000/SUNW,emlxs@2,1/fp@0,0:devctl (NOT CONNECTED)

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

You must choose from one of the available HBAs in the list by entering the appropriate number. In this example, if you enter 2, the utility displays the HBA device name selected and presents a list of command options:

```
HBA: /devices/pci@1e,600000/SUNW,emlxs@2/fp@0,0:devctl  
Available commands:
```

```
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 parameters 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 Leadville state of a specified device.  
dev_remove <wwpn> - Remove the FC device from Leadville management.  
link_status <d id> - Request link error status from a specified D_ID.  
get_fcode_rev - Returns the current Fcode revision of the HBA.  
download_fcode [filename] - Download the HBA fcode.  
get_fw_rev - Returns the current firmware revision of the HBA.  
download_fw [filename] - Download the HBA firmware.  
get_boot_rev - Returns the current boot revision 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 <-t,-b> <file> - Saves firmware core dump to a file.  
get_topology - Returns the current FC network topology.
```

```
reset_link <wwpn,0> - Resets the link of a specified FC device.
reset_hard - Reset the HBA.
reset_hard_core - Reset the HBA firmware core.
diag <test> - 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 or <number> [-i interval] - Returns the driver's internal message log.
get_host_attrs - Returns the host adapter and port attributes.
get_port_attrs <index>, <wwpn> or all - Returns the port attributes.
get_path <index> - Returns the adapter path.
q - Exits this program.
h - Returns this help screen.
hba - Select another hba.
p - Repeat previous command.
emlxadm>
```

At the bottom of the command list is an **emlxadm>** prompt. From this point, the utility is prompt driven. When the prompt is displayed, you must enter one of the commands in the list. The list is displayed automatically only once, but you can display it again by entering **h** at the prompt. To exit the program, enter **q**.

Some commands require additional arguments, such as a Fibre Channel World Wide Port Name (WWPN) or a Fibre Channel port address (D_ID). To display the available arguments for a command, enter the command without any arguments.

For example, the command **get_state** requires a WWPN for the target device. If only the command without the argument is entered, the following statement appears to indicate that the command requires an argument to be executed. For example:

```
emlxadm> get_state
Usage: get_state <wwpn>
emlxadm> get_state 21000020371938fa
State: PORT_DEVICE_LOGGED_IN
```

For a detailed explanation of each command and its arguments, see *Command Descriptions (emlxadm)* on page 8.

CLI Mode (emlxadm)

The emlxadm utility program can be run in CLI mode by typing the name of the program, followed by the full device name of the desired HBA, followed by a valid command and any required command arguments. For example:

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

This mode of operation enables you to use the emlxadm utility as part of a script or another program capable of executing system level calls. For a detailed explanation of each command and its arguments, see *Command Descriptions (emlxadm)* on page 8.

Command Descriptions (emlxadm)

This section provides a list of commands that can be issued with the emlxadm utility program.

get_num_devs

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

Example:

```
emlxadm> get_num_devs

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

get_dev_list

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

Example:

```
emlxadm> get_dev_list

-----
Device 0:
  Dtype: 0
FC4_type[proto]: 0x0100, 0, 0, 0, 0, 0, 0, 0, 0
  State: PORT_DEVICE_LOGGED_IN
  D_id: 113e1
  LILP: 0
  Hard Addr: e1
  WWPN: 21000020371938fa
  WWNN: 20000020371938fa

-----
Device 1:
  Dtype: 0
FC4_type[proto]: 0x0100, 0, 0, 0, 0, 0, 0, 0, 0
  State: PORT_DEVICE_LOGGED_IN
  D_id: 113e2
  LILP: 0
  Hard Addr: e2
  WWPN: 21000020371939a2
  WWNN: 20000020371939a2

-----
Device 2:
  Dtype: 0
FC4_type[proto]: 0x0100, 0, 0, 0, 0, 0, 0, 0, 0
  State: PORT_DEVICE_LOGGED_IN
  D_id: 113e4
  LILP: 0
  Hard Addr: e4
  WWPN: 21000020371938a3
  WWNN: 20000020371938a3

-----
Device 3:
  Dtype: 0
FC4_type[proto]: 0x0100, 0, 0, 0, 0, 0, 0, 0, 0
  State: PORT_DEVICE_LOGGED_IN
  D_id: 113e8
  LILP: 0
  Hard Addr: e8
  WWPN: 2100002037193670
  WWNN: 2000002037193670
```

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
00 00 00 00
```

get_host_params

Returns the FC login parameters of this HBA port.

Example:

```
emlxadm> get_host_params

Host:
      Dtype: 0
FC4_type[proto]: 0x0120, 0, 0, 0, 0, 0, 0, 0
      State: FC_STATE_LOOP
Linkspeed: FC_STATE_2GBIT_SPEED
      D_id: 1
      LILP: 1
Hard Addr: 0
      WWPN: 10000000c93e4be4
      WWNN: 20000000c93e4be4
```

get_sym_pname

Returns the symbolic FC port name of the HBA port. This operation is currently not supported by the Solaris Leadville stack.

Example:

```
emlxadm> get_sym_pname

ioctl: FCIO_GET_SYM_PNAME: Operation not supported
```

set_sym_pname <"string">

Sets the symbolic FC port name of the HBA to the string provided. This operation is currently not supported by the Solaris Leadville stack.

Example:

```
emlxadm> set_sym_pname "Emulex Corporation"

ioctl: FCIO_SET_SYM_PNAME: Operation not supported
```

get_sym_nname

Returns the symbolic FC node name of the HBA port. This operation is currently not supported by the Solaris Leadville stack.

Example:

```
emlxadm> get_sym_nname

ioctl: FCIO_GET_SYM_NNAME: Operation not supported
```

set_sym_nname <"string">

Sets the symbolic FC node name of the HBA to the string provided. This operation is currently not supported by the Solaris Leadville stack.

Example:

```
emlxadm> set_sym_nname "Emulex Corporation"

ioctl: FCIO_SET_SYM_NNAME: Operation not supported
```

dev_login <wwpn>

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

Example:

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

dev_logout

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

Example:

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

get_state <wwpn>

Returns the current Leadville state of the specified FC device on the network.

Example:

```
emlxadm> get_state 21000020371938fa
State: PORT_DEVICE_LOGGED_IN
```

dev_remove <wwpn>

Removes the specified FC device from Leadville management.

WARNING: This command is currently not properly supported in the Leadville stack and will cause the host operating system to panic.

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)
```

get_fcode_rev

Returns the current Fcode revision of the HBA.

Example:

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


download_fcode <filename>

Downloads the specified FCode image file to the HBA.

Example:

```
emlxadm> download_fcode LP10000DC-S.fcode

Image Components: REL type    size=33848
                  DWC file:   BOOT: version=03841512, 1.50a2

Current: Fcode: 1.50a1
New:     Fcode: 1.50a2    33848 (0x8438) bytes

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

Downloading...

Result: Operation successful.
Done.
```

Note: If the file name is not provided, the program attempts to identify the adapter model, then downloads a default FCode image file, if one is available.

get_fw_rev

Returns the current firmware revision of the HBA.

Example:

```
emlxadm> get_fw_rev

Firmware revision: LP10000DC-S 1.90a4
```

download_fw <filename>

Downloads the specified firmware image file to the HBA.

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: 1.90a3
New:     Firmware: 1.90a4    366712 (0x59878) bytes

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

Downloading...

Done.
```

Note: If the file name is not provided, the program attempts to identify the adapter model, then downloads a default firmware image file, if one is available.

get_boot_rev

Returns the current boot revision of the HBA.

Example:

```
emlxadm> get_boot_rev
Firmware revision: LP10000DC-S 5.01a4
```

download_boot <filename>

Downloads the specified boot image file to the HBA.

Example:

```
emlxadm> download_boot TL501A4.PRG
Image Components: REL type    size=143416
                  DWC file:   BOOT: version=03845054, 5.01a4

Current: Boot: 5.01a2
New:     Boot: 5.01a4    143416 (0x23038) bytes

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

Downloading...

Done.
```

Note: If the file name is not provided, the program attempts to identify the adapter model, then downloads a default boot image file, if one is available.

get_dump_size

Returns the byte size of the HBA's firmware core dump buffer.

Example:

```
emlxadm> get_dump_size
Size: 256 (0x100) bytes
```

force_dump

Forces the HBA to perform a firmware core dump to the core dump buffer.

Example:

```
emlxadm> force_dump
Done.
```

get_dump <-t filename.txt or -b filename.bin>

Returns a copy of the HBA's firmware core dump buffer to the specified file in the specified text (-t) or binary (-b) format.

Example:

```
emlxadm> get_dump -t mydump.txt
Done.
```

Following is an example of the text file created by this operation. The binary version of the file has the binary pattern indicated without the column or row labels and white spaces.

```
mydump.txt
          00 01 02 03    04 05 06 07    08 09 0A 0B    0C 0D 0E 0F
-----
00000000: 00 01 02 03    04 05 06 07    08 09 0a 0b    0c 0d 0e 0f
00000010: 10 11 12 13    14 15 16 17    18 19 1a 1b    1c 1d 1e 1f
00000020: 20 21 22 23    24 25 26 27    28 29 2a 2b    2c 2d 2e 2f
```

```

00000030:  30 31 32 33  34 35 36 37  38 39 3a 3b  3c 3d 3e 3f
00000040:  40 41 42 43  44 45 46 47  48 49 4a 4b  4c 4d 4e 4f
00000050:  50 51 52 53  54 55 56 57  58 59 5a 5b  5c 5d 5e 5f
00000060:  60 61 62 63  64 65 66 67  68 69 6a 6b  6c 6d 6e 6f
00000070:  70 71 72 73  74 75 76 77  78 79 7a 7b  7c 7d 7e 7f
00000080:  80 81 82 83  84 85 86 87  88 89 8a 8b  8c 8d 8e 8f
00000090:  90 91 92 93  94 95 96 97  98 99 9a 9b  9c 9d 9e 9f
000000a0:  a0 a1 a2 a3  a4 a5 a6 a7  a8 a9 aa ab  ac ad ae af
000000b0:  b0 b1 b2 b3  b4 b5 b6 b7  b8 b9 ba bb  bc bd be bf
000000c0:  c0 c1 c2 c3  c4 c5 c6 c7  c8 c9 ca cb  cc cd ce cf
000000d0:  d0 d1 d2 d3  d4 d5 d6 d7  d8 d9 da db  dc dd de df
000000e0:  e0 e1 e2 e3  e4 e5 e6 e7  e8 e9 ea eb  ec ed ee ef
000000f0:  f0 f1 f2 f3  f4 f5 f6 f7  f8 f9 fa fb  fc fd fe ff
00000100:

```

get_topology

Returns the FC network topology of the HBA port.

Example:

```

emlxadm> get_topology

Topology: PRIVATE_LOOP

```

reset_link <wwpn or zero for local link>

Resets the local link, if zero is specified, or the link of a specified FC device on the network.

Example:

```

emlxadm> reset_link 0

Done.

```

or

```

emlxadm> reset_link 21000020371938fa

Done.

```

reset_hard

Forces the HBA to perform a hardware reset.

Example:

```

emlxadm> reset_hard

Done.

```

reset_hard_core

Forces the HBA to perform a core firmware reset.

Example:

```

emlxadm> reset_hard_core

Done.

```

diag <test [parameters]> or diag code <cmd_code (hex)>

Performs the specified diagnostics function or command code on the HBA port. This command provides support for the Emulex-specific tests shown below, or generic support to issue an HBA-specific diagnostic code (in hexadecimal) to any third party HBA.

Tests:

```

    emlx_biu [pattern]           - Performs the Bus Interface Unit test.
    emlx_echo <did> [pattern]    - Performs the ECHO test to a specified port id.
    emlx_post                    - Performs the Power-On Self Tests.

```

Parameters:

pattern - 4 byte hex pattern to be used for test. (e.g. 0xA5A5A5A5)

Example:

```
emlxadm> diag emlx_biu
```

```
Result: EMLX_DIAG_BIU: Operation successful.
```

or

```
emlxadm> diag emlx_echo fffffc
```

```
Result: EMLX_DIAG_ECHO: Operation successful.
```

or

```
emlxadm> diag emlx_post
```

```
Result: EMLX_DIAG_POST: Operation successful.
```

Example:

```
emlxadm> diag code 0x4526
```

```
Result: CODE(0x4526): 16 (0x10)
```

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

ns

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

Example:

```
emlxadm> ns
```

```
Nameserver:
```

```

-----
    TYPE: 02
    PID: 0113E1
    WWPN: 21000020371938fa
    PORT_NAME: (SEAGATE ST39103FC          0004)
    WWNN: 20000020371938fa
    NODE_NAME: (null)
    IPA: ffffffffffffffff
    IP_ADDR: 0.0.0.0
    CLASS: 8
    FC4_TYPES:
    00000100,00000000,00000000,00000000,00000000,00000000,00000000,00000000

```

```

-----
    TYPE: 02
    PID: 0113E2
    WWPN: 21000020371939a2
    PORT_NAME: (SEAGATE ST39103FC          0004)
    WWNN: 20000020371939a2
    NODE_NAME: (null)
    IPA: ffffffffffffffff
    IP_ADDR: 0.0.0.0
    CLASS: 8
    FC4_TYPES:
    00000100,00000000,00000000,00000000,00000000,00000000,00000000,00000000

```

```

-----
      TYPE: 02
      PID: 0113E4
      WWPN: 21000020371938a3
PORT_NAME: (SEAGATE ST39103FC      0004)
      WWNN: 20000020371938a3
NODE_NAME: (null)
      IPA: ffffffffffffffff
      IP_ADDR: 0.0.0.0
      CLASS: 8
FC4_TYPES:
00000100,00000000,00000000,00000000,00000000,00000000,00000000,00000000

```

```

-----
      TYPE: 02
      PID: 0113E8
      WWPN: 2100002037193670
PORT_NAME: (SEAGATE ST39103FC      0004)
      WWNN: 2000002037193670
NODE_NAME: (null)
      IPA: ffffffffffffffff
      IP_ADDR: 0.0.0.0
      CLASS: 8
FC4_TYPES:
00000100,00000000,00000000,00000000,00000000,00000000,00000000,00000000

```

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_list

Returns a list of configurable parameters.

Example:

```
emlxadm> parm_get_list
```

Parameter:

```

-----
label: console-notice
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_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_set <label> <value>

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

Example: 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.
```

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

Example: This example attempts to set a static parameter:

```
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.      **
```

msgbuf all or <number> [-i interval]

Displays all or part (the last <number> of lines) of the current driver message log, and can update the screen every <interval> seconds if desired. To stop the program from updating the screen, press <Ctrl+C> to break. If no interval is provided, the current message log is displayed with no additional updates, and the emlxadm prompt returns.

Example:

```
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])
```



```
155130.03: 1002042:[4.00C9]emlxs0: DEBUG: 801: ELS comp. (GA_NXT: CT_ACC:
Rsn=0 Exp=0 [01011500,210000e0])
```

get_host_attrs

Displays all of the current host HBA API attributes.

Example:

```
emlxadm> get_host_attrs

Host Attributes:

Manufacturer           = Emulex
Serial Number          = BG42434343
Model                  = LP10000DC-S
Model Description      = EMULEX LIGHTPULSE LP10000DC-S 2GB PCI-X FIBRE
CHANNEL ADAPTER
Node WWN                = 20000000C9409ED5
Node Symbolic Name     = none
Hardware Version       = 1001206d
Driver Version         = 2.00j.t4 (2005.01.18.16.38)
Optional ROM Version   = 1.50a2
Firmware Version       = 1.90a4
Vendor Specific ID     = 10df
Number of HBA ports    = 2
Driver Name            = Emulex-Sun s10-64 sparcs v2.00j.t4
Last Change            = 1
fp Instance            = 1
Node WWN                = 20000000C9409ED5
Port WWN                = 10000000C9409ED5
Port Fc Id             = 011000
Port Type               = Nport
Port State              = Online
Port Supported COS     = Class 3
Port Supported Fc4 Types:
    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, 00, 00
Port Active Fc4 Types:
    00, 00, 01, 20, 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
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 = 5
```

get_port_attrs <index>, <wwn> or all

Displays the current HBA API port attributes. All of the port 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            = 1
fp Instance            = 1
Node WWN                = 20000000C9409ED5
Port WWN                = 10000000C9409ED5
```

```

Port Fc Id           = 011000
Port Type            = Nport
Port State           = Online
Port Supported COS   = Class 3
Port Supported Fc4 Types:
    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
Port Active Fc4 Types:
    00, 00, 01, 20, 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
Port Symbolic Name   = none
Port Supported Speed = 1Gb 2Gb
Port Speed           = 1Gb
Port Max Frame Size  = 0x800 bytes
Fabric Name          = 000000000000000000
Number of Discovered Ports = 5

```

Port[0] Attributes:

```

Node WWN             = 20000020371938FA
Port WWN             = 21000020371938FA
Port Fc Id           = 0113e1
Port Type            = Unknown
Port State           = Unknown
Port Supported COS   = Unknown
Port Supported Fc4 Types:
    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
Port Active Fc4 Types:
    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
Port Symbolic Name   = SEAGATE ST39103FC      0004
Port Supported Speed = Unknown
Port Speed           = Unknown
Port Max Frame Size  = 0x0 bytes
Fabric Name          = 000000000000000000

```

Port[1] Attributes:

```

Node WWN             = 20000020371938A3
Port WWN             = 21000020371938A3
Port Fc Id           = 0113e4
Port Type            = Unknown
Port State           = Unknown
Port Supported COS   = Unknown
Port Supported Fc4 Types:
    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
Port Active Fc4 Types:
    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
Port Symbolic Name   = SEAGATE ST39103FC      0004
Port Supported Speed = Unknown
Port Speed           = Unknown
Port Max Frame Size  = 0x0 bytes
Fabric Name          = 000000000000000000

```

Port[2] Attributes:

```

Node WWN                = 20000020371939A2
Port WWN                = 21000020371939A2
Port Fc Id              = 0113e2
Port Type               = Unknown
Port State              = Unknown
Port Supported COS      = Unknown
Port Supported Fc4 Types:
    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
Port Active Fc4 Types:
    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
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                = 200000E08B17006F
Port WWN                = 210000E08B17006F
Port Fc Id              = 011500
Port Type               = Unknown
Port State              = Unknown
Port Supported COS      = Unknown
Port Supported Fc4 Types:
    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
Port Active Fc4 Types:
    00, 00, 01, 20, 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
Port Symbolic Name      = none
Port Supported Speed    = Unknown
Port Speed              = Unknown
Port Max Frame Size     = 0x0 bytes
Fabric Name             = 0000000000000000

```

Port[4] Attributes:

```

Node WWN                = 2000002037193670
Port WWN                = 2100002037193670
Port Fc Id              = 0113e8
Port Type               = Unknown
Port State              = Unknown
Port Supported COS      = Unknown
Port Supported Fc4 Types:
    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
Port Active Fc4 Types:
    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
Port Symbolic Name      = SEAGATE ST39103FC      0004
Port Supported Speed    = Unknown
Port Speed              = Unknown
Port Max Frame Size     = 0x0 bytes

```

Fabric Name = 0000000000000000

get_path <index>

Displays the current Solaris device path for a specified HBA 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.

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
```

q

Exits the utility program.

Example:

```
emlxadm> q
Exiting...
```

h

Displays a help menu of utility commands.

Example:

```
emlxadm> h
Available commands:
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 parameters 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 Leadville state of a specified device.
dev_remove <wwpn> - Remove the FC device from Leadville management.
link_status <d_id> - Request link error status from a specified D_ID.
get_fcode_rev - Returns the current Fcode revision of the HBA.
download_fcode [filename] - Download the HBA fcode.
get_fw_rev - Returns the current firmware revision of the HBA.
download_fw [filename] - Download the HBA firmware.
get_boot_rev - Returns the current boot revision 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 <-t,-b> <file> - Saves firmware core dump to a file.
get_topology - Returns the current FC network topology.
reset_link <wwpn,0> - Resets the link of a specified FC device.
reset_hard - Reset the HBA.
reset_hard_core - Reset the HBA firmware core.
diag <test> - 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 or <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> or all - Returns the port attributes.
get_path <index> - Returns the adapter path.
q - Exits this program.
h - Returns this help screen.
hba - Select another hba.
p - Repeat previous command.

hba

Allows you to select another HBA to interface with. This prevents you from having to exit and reenter the program.

Example:

```
emlxadm> hba

Available HBA's:

1. /devices/pci@1e,600000/SUNW,qlc@3/fp@0,0:devctl (CONNECTED)
2. /devices/pci@1e,600000/SUNW,qlc@3,1/fp@0,0:devctl (NOT CONNECTED)
3. /devices/pci@1e,600000/SUNW,emlxs@2/fp@0,0:devctl (CONNECTED)
4. /devices/pci@1e,600000/SUNW,emlxs@2,1/fp@0,0:devctl (NOT CONNECTED)

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

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.
```

Using the emlxdrv Utility

The emlxdrv utility is intended to be used for binding (associating) the Emulex emlxs (Leadville Fibre Channel) driver and the Emulex lpfc (traditional non-Leadville Fibre Channel) driver to the various models of Emulex Fibre Channel HBAs. This allows both drivers to coexist in the same host and attach to mutually exclusive Emulex Fibre Channel HBA models. In other words, the emlxs driver can be configured to attach and operate one set of HBA models, while the lpfc driver can be configured to attach and operate a different set of HBA models. However, Solaris does not allow both drivers to attach and operate the same model of HBA even if there are multiple HBAs of that model present. If the driver binding configuration is changed, the host system must usually be rebooted in order for the new configuration to take effect.

Modes of Operation (emlxdrv)

The emlxdrv utility program can be run in two modes:

- Interactive
- CLI

Interactive Mode (emlxdrv)

Run the emlxdrv utility program in interactive mode by typing the name of the program without any command line arguments:

```
# emlxdrv
```

After it is started, the emlxdrv program scans the host system and prepares a driver configuration table consisting of bindings (associations) between the emlxs and lpfc drivers and a list of Emulex Fibre Channel HBA models. After the table is prepared, the utility displays the following:

```
EMLXDRV Driver Management Utility, Version 1.00j
COPYRIGHT © 2004-2005 Emulex. All rights reserved.
```

```
Driver Alias Present Boot Sun Models
-----
-      lpfs  no      no      no  LP8000S and LP9002S (SBUS)
-      f800  no      no      no  LP8000 and LP8000DC
lpfc   f900  yes     no      no  LP9002, LP9002C, LP9002DC, and LP9402DC
lpfc   f980  no      no      no  LP9802 and LP9802DC
emlxs  fa00  yes     no      no  LP10000, LP10000DC and LP10000ExDC
emlxs  fd00  no      no      no  LP11000 and LP11002
emlxs  fe00  no      no      no  LPe11000 and LPe11002
emlxs  f0a5  no      no      no  2G Blade Adapter (emlxs only)
emlxs  fc00  yes     no      yes  LP10000-S and LP10000DC-S
emlxs  fc20  no      no      yes  LPe11000-S and LPe11002-S
```

```
Available commands:
```

```
-----
set_emlxs <Alias> - Sets emlxs driver to bind to the specified device(s)
set_emlxs_sun     - Sets emlxs driver to bind to all Sun devices
set_emlxs_all     - Sets emlxs driver to bind to all devices
set_lpfc <Alias>  - Sets lpfc driver to bind to the specified device(s)
set_lpfc_nonsun  - Sets lpfc driver to bind to all non-Sun devices
clear_dev <Alias> - Clears driver binding to the specified device(s)
clear_lpfc        - Clears all lpfc driver bindings
clear_emlxs       - Clears all emlxs driver bindings
clear_sun         - Clears driver bindings to all Sun devices
clear_nonsun      - Clears driver bindings to all non-Sun devices
clear_all         - Clears driver bindings to all devices
q                 - Exits this program.
```

```
emlxdrv>
```

The display comprises three parts: the current driver configuration table, a list of available commands and the emlxdrv prompt.

The driver configuration table contains the following columns of data:

- **Driver:** Indicates which driver (emlxs, lpfc or "-" if none) is currently configured to bind or attach to a specific adapter alias.
- **Alias:** Indicates the specific adapter alias associated with a set of Emulex HBA models. Driver bindings can be made only with a specific adapter alias and not with a specific adapter model.
- **Present:** Indicates whether this specific type of adapter is currently present in the host system. Emlxdrv allows you to bind a driver to adapters that are not currently present in the system but that may be present at some point in the future.
- **Boot:** Indicates whether this specific type of adapter is currently providing connectivity to the system's boot disk. This is important because emlxdrv 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.
- **Sun:** Indicates whether this specific type of adapter is branded and sold directly by Sun Microsystems.
- **Models:** Provides a list of Emulex HBA 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.

After the driver configuration table is a list of available commands. For a detailed explanation of each command and its arguments, see *Command Descriptions (emlxdrv)* on page 28.

Below the command list is an **emlxdrv>** prompt. From this point, the utility is prompt driven. When the prompt is displayed, you must enter one of the commands in the list. The current driver configuration table and the available command list are displayed automatically after each command is issued.

Some commands require an additional <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.

To exit the program, enter **q**.

CLI Mode (emlxdrv)

The emlxdrv utility program can be run in CLI mode by typing the name of the program followed by a valid command and any required command arguments. For example, you can update the a device binding by entering all the information on one line at the operating system prompt:

```
# emlxdrv set_emlxs f980
Updating f980 ...
Done.

Driver Alias Present Boot Sun Models
-----
emlxs lpfs no no no LP8000S and LP9002S (SBUS)
- f800 no no no LP8000 and LP8000DC
lpfc f900 yes no no LP9002, LP9002C, LP9002DC, and LP9402DC
lpfc f980 no no no LP9802 and LP9802DC
emlxs fa00 yes no no LP10000, LP10000DC and LP10000ExDC
emlxs fd00 no no no LP11000 and LP11002
emlxs fe00 no no no LPe11000 and LPe11002
emlxs f0a5 no no no 2G Blade Adapter (emlxs only)
emlxs fc00 yes no yes LP10000-S and LP10000DC-S
emlxs fc20 no no yes LPe11000-S and LPe11002-S
```

#

This mode of operation enables you to use the `emlxdrv` utility as part of a script or another program capable of executing system-level calls. For a detailed explanation of each command and its arguments, see *Command Descriptions (emlxdrv)* on page 28.

Command Descriptions (emlxdrv)

This section provides a list of commands that can be issued with the `emlxdrv` utility program. You can view the list of commands at any time by running the `emlxdrv` utility in interactive mode (see *Interactive Mode (emlxdrv)* on page 26).

set_emlxs <alias>

Sets the `emlxs` driver to bind to the specified devices. You must specify one of the valid adapter aliases listed on the screen. 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.

You may see the message "Cannot unload module". This indicates that you must reboot the system to get a driver to unbind from that adapter alias; `emlxdrv` only updates the system configuration for the next boot.

Example:

```
emlxdrv> set_emlxs f980

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

Done.
```

set_emlxs_sun

Sets the `emlxs` driver to bind to all Sun devices.

Example:

```
emlxdrv> set_emlxs_sun

Updating fc00 ...
Done.
```

set_emlxs_all

Sets the `emlxs` driver to bind to all devices. You may see the message "Cannot unload module". This indicates that you must reboot the system to get a driver to unbind from that adapter alias; `emlxdrv` only updates the system configuration for the next boot.

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 ...
Done.
```

set_lpfc <alias>

Sets the lpfc driver to bind to the specified devices. You must specify one of the valid adapter aliases listed on the screen. 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.

You may see the message "Cannot unload module". This indicates that you must reboot the system to get a driver to unbind from that adapter alias; emlxdrv only updates the system configuration for the next boot.

Example:

```
emlxdrv> set_lpfc fa00

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

set_lpfc_nonsun

Sets the lpfc driver to bind to all non-Sun devices. You may see the message "Cannot unload module". This indicates that you must reboot the system to get a driver to unbind from that adapter alias; emlxdrv only updates the system configuration for the next boot.

Example:

```
emlxdrv> set_lpfc_nonsun

Updating lpfs ...
Updating f800 ...
Updating f900 ...
Updating f980 ...
Updating fa00 ...
Cannot unload module: emlxs
Will be unloaded upon reboot.

Updating fd00 ...
Updating fe00 ...
Done.
```

clear_dev <alias>

Clears driver binding to the specified devices. You must specify one of the adapter aliases listed on the screen. 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.

You may see the message "Cannot unload module". This indicates that you must reboot the system to get a driver to unbind from that adapter alias; emlxdrv only updates the system configuration for the next boot.

Example:

```
emlxdrv> clear_dev fe00

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

clear_lpfc

Clears all lpfc driver bindings. You may see the message "Cannot unload module". This indicates that you must reboot the system to get a driver to unbind from that adapter alias; emlxdrv only updates the system configuration for the next boot.

Example:

```
emlxdrv> clear_lpfc

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

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

Done.
```

clear_emlxs

Clears all emlxs driver bindings. You may see the message "Cannot unload module". This indicates that you must reboot the system to get a driver to unbind from that adapter alias; emlxdrv only updates the system configuration for the next boot.

Example:

```
emlxdrv> clear_emlxs

Cannot unload module: emlxs
Will be unloaded upon reboot.

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

Done.
```

clear_sun

Clears driver bindings to all Sun devices. You may see the message "Cannot unload module". This indicates that you must reboot the system to get a driver to unbind from that adapter alias; emlxdrv only updates the system configuration for the next boot.

Example:

```
emlxdrv> clear_sun

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

Done.
```

clear_nonsun

Clears driver bindings to all non-Sun devices. You may see the message "Cannot unload module". This indicates that you must reboot the system to get a driver to unbind from that adapter alias; emlxdrv only updates the system configuration for the next boot.

Example:

```
emlxdrv> clear_nonsun

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_all

Clears driver bindings to all devices. You may see the message "Cannot unload module". This indicates that you must reboot the system to get a driver to unbind from that adapter alias; emlxdrv only updates the system configuration for the next boot.

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.

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

Done .

q

Exits the program. If changes were made to the driver bindings, a system reboot is usually required in order for all the changes to take effect.

Example:

```
emlxdrv> q
```

```
Exiting...
```

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

```
#
```