FCA Utilities

Version 1.03h

User Manual

for the

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

P003594-01A Rev. A

One Network.
One Company.
Connect with Emulex.

Emulex, 3333 Susan Street
Costa Mesa, CA 92626

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 (FC).

New in this Release

The following commands are new to the 1.03h version:

- Force_dump. This command forces the adapter to perform a firmware core dump to the core dump buffer.
- Get_dump. This command saves the firmware core dump to a file.
- Get_dump_size. This command returns the adapter's firmware core dump size.

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 8 gigabits per second, or roughly 800 megabytes per second. FC 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 FC specifications provide for the emulation of two traditional protocols: SCSI and IP. For storage, FC 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, FC provides an encapsulation of IP (Internet Protocol), referred to in this document as *IP*.

The Solaris Fibre Channel Stack

Each FC adapter 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® adapter 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.

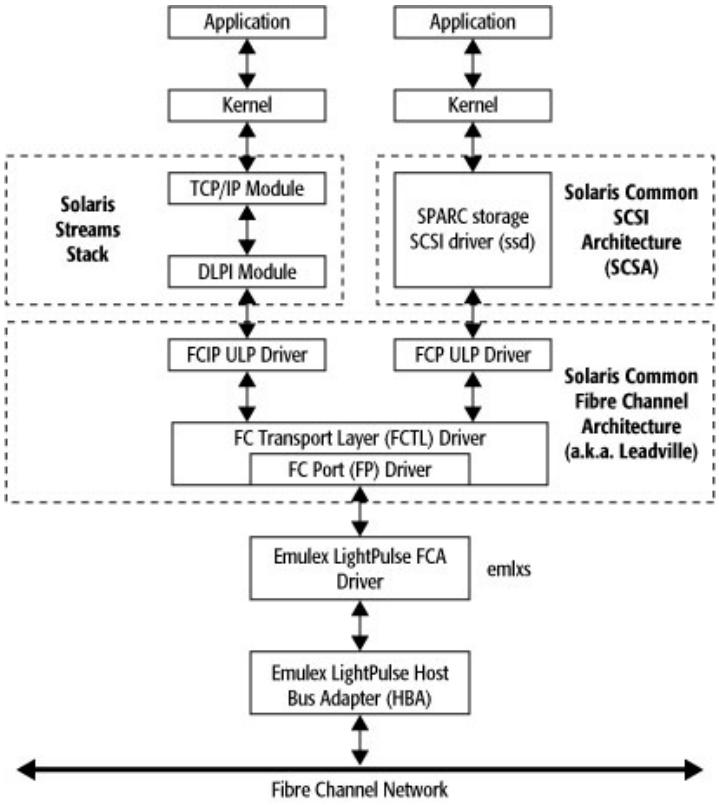

Figure 1: The Solaris Software Stack

Installing the Utilities

The Emulex utility programs are provided on the Emulex Web site. You can download them by going to http://www.emulex.com/support/solaris/index.jsp and selecting the appropriate Solaris SFS operating system version. Then look for **Utilities** and click **Download**.

Compatibility

The StorEdge SAN Foundation Software (SFS) driver and utilities support the following operating systems:

- Solaris 10 SPARC
- Solaris 10 x64 and x86

The following table specifies the adapters supported by the Solaris SFS FCA (emlxs) driver and the Emulex Fibre Channel Adapter Utilities (FCA Utilities).

Table 1: Adapter Compatibility

			FCA Utilities	
Adapter	**Solaris SFS Driver**	**HBAnyware® Utility**	**emlxadm**	**emlxdrv**
SG-XPCIE20FC-NEM-Z*	X	X	X	N/A
SG-XPCIE2FC-ATCA-Z*	X	X	X	N/A
SG-XPCI1FC-EM2*	X	X	X	N/A
SG-XPCI2FC-EM2*	X	X	X	N/A
SG-XPCIE2FC-EB4-Z	X	X	X	X
LP21002	X	X	**	**
LP21000	X	X	**	**
LPe12002	X	X	X	X
LPe12000	X	X	X	X
LP11002	X	X	X	X
LP11000	X	X	X	X
LPe11004	X	X	X	X
LPe11002	X	X	X	X
LPe11000	X	X	X	X
LP10000ExDC	X	X	X	X
LP10000DC	X	X	X	X
LP10000	X	X	X	X
LP9802	X	X	X	X
LP9002DC	X	X	X	X
LP9002L	X	X	X	X
LP9002S	X	X	X	X
** These utilities support the Fibre Channel (FC) component of converged network adapters (CNAs) such as the LP21000 and LP21002. These utilities do not support the Ethernet component of CNAs.				

* Sun branded adapter. Special firmware installation considerations apply. Refer to the HBAnyware User Manual for more information about updating firmware.

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 completely install 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:

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 shows 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 5 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 need to reboot your system to run 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 7. If you are updating the emlxu utilities to a newer version and you have the new utilities kit tar file, you do not need 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 4 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 shows 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 4.
- “Installing the Utilities Package Manually” on page 6.

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 7, 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:

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 need to reboot your system to run 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:

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
   ```

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

To update the emlxu utilities:

1. Remove the old emlxu utilities package by following the instructions in “Removing the Utilities Package Manually” on page 7.
2. Install the new emlxu utilities package by following the instructions in “Installing the Utilities Package Manually” on page 6.

Using the emlxadm Utility

The emlxadm utility is used to change driver parameters through a local interactive or command-line interface. It can also be used for firmware updates on non-Sun branded devices. HBAnyware 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. It 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 driver, which manages the FCA drivers for each Fibre Channel adapter 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. For example:

```
# emlxadm
```

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

```
EMLXADM Device Management Utility, Version: v1.03h
COPYRIGHT (c) 2004-2009 Emulex. All rights reserved.

Available Emulex HBAs:

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:
```

Enter an adapter number or zero to exit: You must choose from one of the available adapters in the list by entering the appropriate number. In this example, if you enter ***<1>***, the utility shows the adapter device name selected and presents a list of command options:

```
HBA 1: /devices/pci@7c0/pci@0/pci@1/pci@0,2/SUNW,emlxs@1/fp@0,0

Available commands:
[FCIO rev2]

get_num_dev - 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 SFS state of a specified device.
dev_remove <wwpn> - Removes the FC device from SFS management.
link_status <d_id> - Requests link error status from a specified D_ID.
get_fcode_rev - Returns the current Fcode revison of the HBA.
download_fcode [filename] - Downloads the HBA fcode.
get_fw_rev - Returns the current firmware revison of the HBA.
download_fw [filename] - Downloads the HBA firmware.
get_boot_rev - Returns the current boot revison of the HBA.
download_boot [filename] - Downloads the HBA boot image.
get_dump_size - Returns the HBA's firmware core dump size.
force_dump - Forces 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 FC device.
reset_hard - Resets the HBA.
diag - Performs 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.
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 typing **<h>** at the prompt. To exit the program, type **<q>**.

Some commands require additional arguments, such as a FC World Wide Port Name (WWPN) or a FC 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
```

CLI Mode (emlxadm)

There are two ways to run emlxadm in a CLI mode:

1. Device path option:

```
USAGE:  emlxadm <device path> [-y] <cmd> [cmd_options]
```

Run the emlxadm utility program using the device path option by typing the name of the program, followed by the full device name of the desired adapter (or a pattern string for multiple adapters), followed by a valid command and any required command arguments. In the following example, the emlxadm utility pauses to ask if you want to continue before executing the command. To specify a full device name, type:

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

Information similar to the following is displayed:

```
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 <---Response
required

State: PORT_DEVICE_LOGGED_IN
#
```

If you do not want the utility to pause for verification, add a "-y" option just after the device path, and the emlxadm utility will skip the verification. For example:

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

Information similar to the following is displayed:

```
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

#
```

If you want to run a command on multiple adapters at once, you can use a pattern string instead of a full device path. If the entire pattern string matches any part of an adapter device path, the command will execute against that adapter. Again, in this example the emlxadm utility pauses to ask if you want to continue before executing the command. For example:

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

Information similar to the following is displayed:

```
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   <-- (Response
required)

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   <-- (Response
required)

There are 0 devices reported on this port.

#
```

If you do not want the utility to pause for verification, add a "-y" option just after the pattern string, and the emlxadm utility will skip the verification. For example:

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

Information similar to the following is displayed:

```
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.

#
```

2. Instance option:

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

Run the emlxadm utility program using the instance option by typing the name of the program, followed by one of the options listed below, followed by a valid command and any required command arguments.

```
 <N>  - A specific emlxs driver instance. (N=1 means emlxs1, N=2 means emlxs2).
SFS   - All emlxs driver instances attached to the Sun SFS interface.
FCT   - All emlxs driver instances attached to the Sun COMSTAR. interface.
all   - All emlxs driver instances.
```

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.

Command Descriptions (emlxadm)

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

Table 2: Alphabetical Listing of All Commands

Command	Description	For an example, see
boot_code [enable or disable]	Sets or shows the boot code state of the current adapter.	page 14
dev_login <wwpn>.	Performs an FC login to an FC device on the network, if not already logged in.	page 14
dev_logout <wwpn>	Performs an FC logout to an FC device on the network, if not already logged out.	page 14
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.	page 14
diag <test [parameters]> or diag code <cmd_code (hex)>	Performs the specified diagnostics function or command code on the adapter port. This command provides support for the Emulex-specific tests shown below, or generic support to issue an adapter specific diagnostic code (in hexadecimal) to any third party adapter.	page 15

Table 2: Alphabetical Listing of All Commands (Continued)

Command	Description	For an example, see
download_boot <filename>	Downloads the specified boot image file to the adapter.	page 15
download_fcode <filename>	Downloads the specified FCode image file to the adapter.	page 16
download_fw <filename>	Downloads the specified firmware image file to the adapter.	page 16
force_dump	Forces the adapter to perform a firmware core dump to the core dump buffer.	page 16
get_boot_rev	Returns the current boot revision of the adapter.	page 16
get_dev_list	Returns a list of FC devices currently seen by this adapter port.	page 17
get_dump	Saves the firmware core dump to a file.	page 17
get_dump_size	Returns the adapter's firmware core dump size.	page 18
get_fcode_rev	Returns the current FCode revision of the adapter.	page 18
get_fw_rev	Returns the current firmware revision of the adapter.	page 18
get_host_attrs	Returns all of the current host adapter API attributes.	page 18
get_host_params	Returns the FC login parameters of this adapter port.	page 19
get_inst	Returns the driver instance.	page 19
get_logi_params <wwpn>	Returns the FC login common service parameters for a specified FC device on the network.	page 19
get_num_devs	Returns the number of FC devices currently seen by this adapter port.	page 16
get_path <index>	Returns 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.	page 20
get_port_attrs <index>, <wwn> or all	Returns the current adapter 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.	page 20

Table 2: Alphabetical Listing of All Commands (Continued)

Command	Description	For an example, see
get_rnid [wwpn]	Gets the RNID information for the local or specified port.	page 22
get_state <wwpn>	Returns the current Leadville state of the specified FC device on the network.	page 22
get_sym_nname	Returns the symbolic FC node name of the adapter port. **Note:** This operation is currently not supported by the Solaris Leadville stack.	page 22
get_sym_pname	Returns the symbolic FC port name of the adapter port. **Note:** This operation is currently not supported by the Solaris Leadville stack.	page 22
get_topology	Returns the FC network topology of the adapter port.	page 23
get_vpd	Returns the current adapter's vital product data (VPD).	page 23
h	Returns a help menu of utility commands.	page 23
hba	Allows you to select another adapter to interface with. This prevents you from having to exit and reenter the program.	page 24
link_status <d_id>	Requests and returns the current link error status from the FC device specified by the D_ID address.	page 24
msgbuf all or <number> [-i interval]	Returns 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>**. If no interval is provided, the current message log is displayed with no additional updates, and the emlxadm prompt returns.	page 25
ns	Performs and returns a complete query of the fabric name server.	page 25
p	Repeats the last command.	page 29
parm_get <label>	Gets the value of a specified parameter in the driver.	page 29
parm_get_list	Returns a list of configurable parameters.	page 26
parm_get_num	Returns the total number of configurable parameters.	page 26

Table 2: Alphabetical Listing of All Commands (Continued)

Command	Description	For an example, see
parm_set <label> <value>	Sets the value of a specified parameter in the driver. Only dynamic parameters can be set.	page 29
q	Exits the utility program.	page 29
reset_hard	Forces the adapter to perform a hardware reset.	page 29
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.	page 30
set_sym_nname <"string">	Sets the symbolic FC node name of the adapter to the string provided. **Note:** This operation is currently not supported by the Solaris Leadville stack.	page 30
set_sym_pname <"string">	Sets the symbolic FC port name of the adapter to the string provided. **Note:** This operation is currently not supported by the Solaris Leadville stack.	page 30

Command Usage

boot_code [enable or disable]

Sets or shows the boot code state of the current adapter.

Example:

```
emlxadm> boot_code
Boot code: Disabled

emlxadm> boot_code enable
Boot code: Enabled

emlxadm> boot_code disable
Boot code: Disabled
```

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 <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 Leadville management.

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

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

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

```
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 adapter is displayed in decimal and hexadecimal if the diagnostic code is valid for the adapter. No interpretation of the return status is provided.

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: 1.50a1
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.
```

download_fw <filename>

Downloads the specified firmware image file to the adapter.

Note: The Emulex FCA driver does not allow firmware updates to SUN 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: 1.90a3
New:     Firmware: 1.90a4  366712 (0x59878) bytes

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

Downloading...
Done.
```

force_dump

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

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 <get_dump or get_dump -h>

Saves firmware core dump to a file. The DMP and TXT files are created in the current working directory. If -h is specified, then the only the header information of the firmware core is displayed. The core files are not created.

Example:

```
emlxadm> get_dump
Core size: 2033060 bytes
       files: 2
  TXT file: 10348
  DMP file: 2022692
```

get_dump_size

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

Example:

```
emlxadm> get_dump_size
Size: 2033060 (0x1f05a4) 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_host_attrs

Returns all of the current host adapter API attributes.

Example:

```
emlxadm> get_host_attrs
Hoxc/'
FD/st Attributes:

    Manufacturer            = Sun 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_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>

Returns 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.

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_port_attrs <index>, <wwn> or all

Returns 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               = 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

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]

Gets the RNID 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 Leadville state of the specified FC device on the network.

Example:

```
emlxadm> get_state 21000020371938fa

State: PORT_DEVICE_LOGGED_IN
```

get_sym_pname

Returns the symbolic FC port name of the adapter port.

Example:

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

get_sym_nname

Returns the symbolic FC node name of the adapter port.

Example:

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

get_topology

Returns the FC network topology of the adapter port.

Example:

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

get_vpd

Returns the current adapter's vital product data (VPD).

Example:

```
emlxadm> get_vpd
Vital Product Data:
     Identifier (ID): FC2G PCI-X LP10000DC - SUN
    Part Number (PN): LP10000DC-S
   Manufacturer (MN): Sun Microsystems, Inc.
  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 a help menu of utility commands.

Example:

```
emlxadm> h

get_num_dev - 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> - Removes the FC device from SFS management.
link_status <d_id> - Requests link error status from a specified D_ID.
get_fcode_rev - Returns the current Fcode revison of the HBA.
download_fcode [filename] - Downloads the HBA fcode.
get_fw_rev - Returns the current firmware revison of the HBA.
download_fw [filename] - Downloads the HBA firmware.
get_boot_rev - Returns the current boot revison of the HBA.
download_boot [filename] - Downloads the HBA boot image.
get_dump_size - Returns the HBA's firmware core dump size.
force_dump - Forces 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 FC device.
reset_hard - Resets the HBA.
diag - Performs 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.
q - Exits this program.
h - Returns this help screen.
hba - Select another hba.
p - Repeat previous command..

emlxadm>
```

hba

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

Example:

```
Available Emulex HBA's:

1. SFS:emlxs0 : /devices/pci@7c0/pci@0/pci@1/pci@0,2/SUNW,emlxs@1/fp@0,0 (CON-
NECTED)
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 or <number> [-i interval]

Returns 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>**. 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])
```

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
```

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.

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.        **
```

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.
```

q

Exits 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 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.
```

set_sym_nname <"string">

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

Note: 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
```

set_sym_pname <"string">

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

Note: 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
```

Using the emlxdrv Utility

The emlxdrv utility is used to bind (associate) 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 adapters. This utility is used for migrating a server and its adapters from LPFC to emlxs or vice versa. Note that the simultaneous use of both drivers is not supported in production and should only be used as a step when migrating complex configurations between drivers. 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 adapter models. After the table is prepared, the utility returns the following:

```
EMLXDRV Driver Management Utility, Version: v1.00m
COPYRIGHT © 2004-2008 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,LPe11002 and LPe11004
 emlxs  f100  no      no   no  LPe12000 and LPe12002
 emlxs  fe05  no      no   no  LP21000 and LP21002
 emlxs  f0a5  no      no   no  2G Blade Adapter (emlxs only)
 emlxs  fc00  yes     no   yes LP10000-S and LP10000DC-S (emlxs only)
 emlxs  fc10  no      no   yes LP11000-S and LP11002-S (emlxs only)
 emlxs  fc20  no      no   yes LPe11000-S and LPe11002-S (emlxs only)
 emlxs  fc40  no      no   yes LPe12000-S and LPe12002-S (emlxs only)
```

```
 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 adapter 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 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.

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 33.

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.

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
-------------------------------------------------------------------------
 -      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
 emlxs  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,LPe11002 and LPe11004
 emlxs  f100  no      no   no  LPe12000 and LPe12002
 emlxs  fe05  no      no   no  LP21000 and LP21002
 emlxs  f0a5  no      no   no  2G Blade Adapter (emlxs only)
 emlxs  fc00  yes     no   yes LP10000-S and LP10000DC-S (emlxs only)
 emlxs  fc10  no      no   yes LP11000-S and LP11002-S (emlxs only)
 emlxs  fc20  no      no   yes LPe11000-S and LPe11002-S (emlxs only)
 emlxs  fc40  no      no   yes LPe12000-S and LPe12002-S (emlxs only)

#
```

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.

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 31).

Table 3: Alphabetical Listing of All Commands

Command	Description	See
clear_all	Clears driver bindings to all devices.	page 34
clear_dev <alias>	Clears driver binding to the specified devices.	page 35
clear_emlxs	Clears all emlxs driver bindings.	page 35
clear_lpfc	Clears all lpfc driver bindings.	page 35
clear_nonsun	Clears driver bindings to all non-Sun devices.	page 36
clear_sun	Clears driver bindings to all Sun devices.	page 36
q	Exits the program.	page 37
set_emlxs <alias>	Sets the emlxs driver to bind to the specified devices.	page 37
set_emlxs_all	Sets the emlxs driver to bind to all devices.	page 37
set_emlxs_sun	Sets the emlxs driver to bind to all Sun devices.	page 38

Table 3: Alphabetical Listing of All Commands (Continued)

Command	Description	See
set_lpfc <alias>	Sets the lpfc driver to bind to the specified devices.	page 38
set_lpfc_nonsun	Sets the lpfc driver to bind to all non-Sun devices.	page 38

Command Usage

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.

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

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

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_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.

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

Updating fc20 ...
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_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_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.

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 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.
#
```

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_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 ...
Updating fc10 ...
Updating fc20 ...
Done.
```

set_emlxs_sun

Sets the emlxs driver to bind to all Sun devices.

Example:

```
emlxdrv>   set_emlxs_sun

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