Emulex Drivers for Linux

User Manual

One Network.
One Company.
Connect with Emulex.

P005239-01A Rev. A

Information furnished by Emulex is believed to be accurate and reliable. However, no responsibility is assumed by Emulex 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, copyright, trade secret or related rights of Emulex.

Emulex provides this documentation “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 may make improvements and changes to the product described in this manual at any time and without any notice. Emulex 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 documentation, Emulex disclaims any undertaking to give notice of such changes.

Emulex, 3333 Susan Street
Costa Mesa, CA 92626

Supported Driver Versions

Driver Version	Inbox	DUP/DUD	Out of Box	OS Versions Supported	Comments
FC and FCoE Drivers					
8.3.5.8.1p	X			SLES 11 SP1	
8.3.5.14.4p		X		SLES 11 SP1	
8.3.5.17	X			RHEL 6	
8.3.5.25			X	SLES 11 SP1	
Ethernet Drivers					
2.102.147s	X			SLES 11 SP1	
2.102.348.0		X		SLES 11 SP1	
2.102.426r	X			RHEL 6	
2.103.269.2			X	SLES 11 SP1	
iSCSI Drivers					
2.102.527.0	X			SLES 11 SP1	
2.102.348.0		X		SLES 11 SP1	
2.0.527.0	X			RHEL 6	
2.103.269.2			X	SLES 11 SP1	

Installation

Driver Information

Supported Features

- SNIA-CTP compliant SMI-S 1.1 Provider
- Topology support: Fibre Channel Arbitrated Loop (FC-AL), point-to-point, fabric with auto-topology negotiation, and Fibre Channel over Ethernet (FCoE).
- Supports 1, 2, 4 and 8 Gb/s capable FC adapters with auto-rate negotiation and 10Gb/s FCoE adapters. (1Gb/s is not supported on 8 Gb/s adapters.)
- Protocols:
    - iSCSI (supported Linux kernel is SLES 11 SP1, supported kernel variants for x86 and x86_64)
    - NIC (supported Linux kernel is SLES 11 SP1, supported kernel variants for x86 and x86_64)
    - SCSI-FCP
    - FCP-2 (FC-Tape profile, including use of ADISC instead of PLOGI)
    - FC initiator mode and FCoE
- Tested up to thirty-two adapter ports
- Dynamic parameter setting using the Emulex OneCommand™ Manager application as part of a master kit: enabling GUI-based driver configuration and persistent binding management, including in-band (FC) and out-of-band (TCP/IP) remote SAN management capability, diagnostics (loopback and diagnostics dump), and virtual port support. See the *OneCommand Manager Application Version 5.1 User Manual* (on the Emulex Web site) for a complete list of supported features.
- Support for common host bus adapter application programming interface (HBA API)
- Batch firmware download capability
- Support for the sysfs interface
- PCI hot plug support
- Vital Product Data (VPD) support
- “Linux Tools” link on the Linux portion of the Emulex Web site (visit the link for available tools)
- Supports NPIV virtual ports

New Features in this Release

- Supports the OneConnect™ OCe10100 Universal Converged Network Adapters (UCNAs).
- Supports iSCSI, NIC, FC initiator, and FCoE protocols.
- Supports the OneCommand Manager application, both GUI and CLI.
- Multiple driver packages are available:
    - Source-based RPM FC and FCoE Driver Kit – This kit supports legacy Emulex HBA adapters and Emulex OneConnect UCNAs and includes the FC and FCoE driver.

- Binary RPM FC and FCoE Driver Kit – This kit supports legacy Emulex HBA adapters and Emulex OneConnect UCNAs and includes the FC and FCoE driver.
- Ethernet Driver Kit – This kit supports Emulex OneConnect UCNAs and includes the Ethernet (NIC) driver.
- iSCSI Driver Kit – This kit supports Emulex OneConnect UCNAs and includes the iSCSI driver.

Prerequisites

To install any of the standalone driver kits, the appropriate distribution kernel development packages must be installed for the currently running kernel, which include the gcc compiler and the kernel sources.

The standalone driver kits support the following distributions:

- Red Hat Enterprise Linux 6 (Intel x86, Intel EM64T, and AMD64 architectures).
- SuSE Linux Enterprise Server 11 SP1 (Intel x86, Intel EM64T, and AMD64 architectures).

Compatibility

For a list of adapters that are compatible with the standalone driver kits, see the specific driver's Downloads page on the Emulex Web site. For compatible firmware versions, see the Downloads page for the specific adapter.

Note: Check the Emulex Web site for the latest firmware releases.

Note: NPIV is supported on Emulex SLI-4 OCe10100 UCNAs. NPIV is also supported on all SLI-3 4 Gb/s and 8 Gb/s adapters. Emulex enterprise class (5 digit adapter model number) and Midrange class (4 digit adapter model number) adapters support SLI-3. Emulex 3 digit model number adapters do not fully support SLI-3 and therefore do not support NPIV. The LPFC 8.3.5.x driver supports all adapters running SLI-2, but NPIV support is not available in SLI-2 mode.

For SLI-4 and SLI-3 supported adapters, use the latest recommended firmware for NPIV support.

Things to Know Before You Download

- You must uninstall any previous driver kits and/or Application Helper Modules that were installed from the Emulex CD or downloaded from the Emulex Web site, (i.e. not part of a distribution), before installing this driver kit.

Known Issues

- See the product release notes for the latest information.

Installing the Source-based RPM FC and FCoE Driver Kit

The lpfc-install script installs the lpfcdriver_2.6 RPM.

The RPM:

- Installs the driver source files to the /usr/src/lpfc directory.
- Builds the driver for the currently running kernel.
- Installs the driver to the proper directory for the currently running kernel. Maintenance and errata kernels are supported.

Once the RPM is installed, the lpfc-install script creates a new ramdisk for the currently running kernel so that the FC and FCoE driver is loaded when the kernel is initialized during system startup.

Note: You must uninstall any previous FC and FCoE driver kits that were installed from the Emulex CD or downloaded from the Emulex Web site, (i.e. not part of a distribution), before installing this driver kit. This installation will fail if a previous version of the FC and FCoE driver kit is detected.

Refer to “Uninstalling the Source-based RPM FC and FCoE Driver Kit” on page 9 for more information.

When invoked without options, the 'lpfc-install' script automatically archives any driver that is shipped as part of the distribution's kernel during the installation procedure. Old drivers that are archived during installation are then restored when the driver kit is uninstalled.

Note: The OneCommand Manager application must be installed separately from the driver. Refer to the *OneCommand Manager Application Version 5.1 User Manual* for more information.

Note: The lpfc-install script does not support custom kernels. For example, kernels with Version_Release strings that do not match those of the original distribution kernel.

To install the source-based RPM FC and FCoE driver for Linux:

1. Install a supported Emulex adapter in the system. Refer to the adapter's installation manual for specific hardware installation instructions.
2. Remove any previously installed FC and FCoE driver kits that were installed from the Emulex CD or downloaded from the Emulex Web site, (i.e. not part of a distribution’s kernel) before proceeding. Refer to “Uninstalling the Source-based RPM FC and FCoE Driver Kit” on page 9 for more information.
3. Download the driver kit from the Emulex Web site or copy it to the system from the installation CD.
4. Log on as 'root' to a terminal, and unpack the tarball with the following command:

   ```
   tar xzf lpfc_2.6-<driver_kit>-<driver version>.tar.gz
   ```
5. Change to the directory that is extracted:

   ```
   cd lpfc_2.6_driver_kit-<driver version>/
   ```
6. Execute the 'lpfc-install' script with no options to install the new driver kit. Type:

   ```
   ./lpfc-install
   ```

   Once the 'lpfc-install' script has completed successfully, the source-based RPM FC and FCoE driver is loaded and Fibre Channel disks that are properly connected to the system are accessible. Reboot the system now to enable the newly added driver options in the ramdisk. You can also reboot the system later if you want.

Source-based RPM FC and FCoE Driver Kit Install Script Options

The following options are available for use with the Emulex install script for the source-based RPM FC and FCoE driver for Linux:

- -h,--help - Prints a help message describing command line parameters.
- -u,--uninstall - Uninstalls the currently installed driver kit.
- --createramdisk - Creates a new ramdisk image. Use this option after you have modified driver parameters in the /etc/modprobe.conf file.

Source-based RPM FC and FCoE Driver Kit Directory Structure

After installation, the following directory is created on the system.

Table 1: Driver Kit Directory Structure

Directory	Description
/usr/src/lpfc	Driver source files.

Installing the Source-based RPM FC and FCoE Driver on Unsupported Linux Distributions

The Emulex version 8.3.5.x drivers for Linux are not intended for, and will not operate on, any kernel prior to 2.6.232. If you are using an earlier 2.6 kernel version, see the Emulex Web site for additional driver configuration, driver version, and operating system support information. To install the Emulex FC and FCoE driver on an unsupported distribution of Linux, refer to the distribution's Web site or http://kernel.org.

Upgrading the Kernel or Applying a Distribution Service Pack or Update

You can install the driver kit into an upgraded kernel. The installation of an update or service pack generally involves updating the kernel.

Installing the Source-based RPM FC and FCoE Driver Kit into an Upgraded Kernel

To install the source-based RPM FC and FCoE driver kit into an upgraded kernel:

1. Execute the lpfc-install script with the '--uninstall' option. Type:
   ```
   /usr/src/lpfc-install --uninstall
   ```
2. Upgrade the kernel and/or distribution.
3. Reboot the system with the new kernel.
4. Download the driver kit from the Emulex Web site or copy it to the system from the installation CD.
5. Log on as 'root' to a terminal, and unpack the tarball with the following command:
   ```
   tar xzf lpfc_2.6_driver_kit-<driver version>.tar.gz
   ```
6. Change to the directory that is extracted:
   ```
   cd lpfc_2.6_driver_kit-<driver version>/
   ```

7. Execute the 'lpfc-install' script with no options to install the new driver kit. Type:

   ```
   ./lpfc-install
   ```

8. Reboot the system to complete re-installation of the Emulex driver.

Installing the Binary RPM FC and FCoE Driver Kit

The binary RPM FC and FCoE driver kit includes the driver that supports the FC and FCoE protocols. The binary RPM FC and FCoE driver kit consists of the following:

- A zipped tar file that includes the driver binary RPMs for a specific driver version and Linux distribution.

  Note: The driver binary RPM packages only support officially-released Linux distribution kernels, and do not support subsequent errata kernels or pre-release distribution kernels. If errata kernel support is required, the standard (source-based RPM) FC and FCoE driver kit should be used instead.

- An installation script, elx_lpfc_install.sh, which installs by default the FC and FCoE driver binary RPM that corresponds to the target system's architecture and kernel memory variant.
- A README file that provides a description of the kit structure, its contents, distribution support scope, and any release notes that apply on this driver version and distribution.
- The driver changelog file.

Once the RPM is installed, the elx_lpfc_install script creates a new ramdisk for the currently running kernel so that the FC and FCoE driver is loaded when the kernel is initialized during system startup.

Note: The OneCommand Manager application must be installed separately from the driver. Refer to the “Installing the OneCommand Manager Application” on page 9 for more information.

To install the binary RPM FC and FCoE driver for Linux:

1. Install a supported Emulex adapter in the system. Refer to the adapter's installation manual for specific hardware installation instructions.
2. Remove any previously installed Ethernet driver kits and/or Application Helper Modules that were installed from the Emulex CD or downloaded from the Emulex Web site, (i.e. not part of a distribution’s kernel) before proceeding. Refer to “Uninstalling the Source-based RPM FC and FCoE Driver Kit” on page 9 for more information.
3. Download the appropriate driver kit from the Emulex Web site or copy it to the system from the installation CD.
4. Log on as 'root' to a terminal, and unpack the tarball with the following command:

   ```
   tar xzf elx-lpfc-dd-<Linux distribution version>-<driver version>.tar.gz
   ```

5. Change to the directory that is extracted:

   ```
   cd elx-lpfc-dd-<Linux distribution version>-<driver version>/
   ```

6. Execute the 'elx_lpfc_install' script with no options to install the new driver kit. Type:

   ```
   ./elx_lpfc_install.sh
   ```

   Once the elx_lpfc_install' script has completed successfully, the Emulex new FC and FCoE driver is loaded and devices that are properly connected to the system are accessible. Reboot the system now to enable the newly added driver options in the ramdisk. You can also reboot the system later if you want.

Uninstalling the Binary RPM FC and FCoE Driver Kit

Note: You must run the uninstall script that shipped with the version of the driver kit you want to remove.

To uninstall the binary RPM FC and FCoE driver:

1. Log on as 'root'.
2. If possible, exit all applications that use Fibre Channel-attached drives, then unmount the drives. If you cannot exit all applications that use Fibre Channel-attached drives, the uninstall will work properly, but you must reboot after the uninstallation is complete.
3. Execute the ‘elx_lpfc_install.sh’ script. with the '--uninstall' option. Type:

```
./elx_lpfc_install.sh --uninstall
```

Installing the Ethernet Driver Kit

The Ethernet driver kit includes the driver that supports the NIC protocol. The Ethernet driver kit consists of the following:

- A zipped tar file that includes the driver binary RPMs for a specific driver version, and for all of the supported Linux distribution kernels.

  Note: The driver binary RPM packages only support officially-released Linux distribution kernels, and do not support subsequent errata kernels or pre-release distribution kernels.

- An installation script, elx_net_install.sh, which installs by default the Ethernet driver binary RPM that corresponds to the target system's architecture and kernel memory variant.
- A README file that provides a description of the kit structure, its contents, and distribution support scope.

Once the RPM is installed, the elx_net_install script creates a new ramdisk for the currently running kernel so that the Ethernet driver is loaded when the kernel is initialized during system startup.

Note: The OneCommand Manager application must be installed separately from the driver. Refer to the “Installing the OneCommand Manager Application” on page 9 for more information.

To install the Ethernet driver for Linux:

1. Install a supported Emulex adapter in the system. Refer to the adapter's installation manual for specific hardware installation instructions.
2. Remove any previously installed Ethernet driver kits and/or Application Helper Modules that were installed from the Emulex CD or downloaded from the Emulex Web site, (i.e. not part of a distribution’s kernel) before proceeding. Refer to “Uninstalling the Source-based RPM FC and FCoE Driver Kit” on page 9 for more information.
3. Download the driver kit from the Emulex Web site or copy it to the system from the installation CD.
4. Log on as 'root' to a terminal, and unpack the tarball with the following command:

```
tar xzf elx-be2net-dd-<driver version>.tar.gz
```

5. Change to the directory that is extracted:

```
cd elx-be2net-dd-<driver version>/
```

6. Execute the 'elx_net_install' script with no options to install the new driver kit. Type:

```
./elx_net_install.sh
```

Once the elx_net_install' script has completed successfully, the Emulex Ethernet driver is loaded and devices that are properly connected to the system are accessible. Reboot the system now to enable the newly added driver options in the ramdisk. You can also reboot the system later if you want.

Uninstalling the Ethernet Driver Kit

Note: You must run the uninstall script that shipped with the version of the driver kit you want to remove.

To uninstall the Ethernet driver:

1. Log on as 'root'.
2. If possible, exit all applications that use Ethernet-attached drives, then unmount the drives. If you cannot exit all applications that use Ethernet-attached drives, the uninstall will work properly, but you must reboot after the uninstallation is complete.
3. Execute the ‘elx_net_install.sh’ script. with the '--uninstall' option. Type:

```
./elx_net_install.sh --uninstall
```

Installing the iSCSI Driver Kit

The iSCSI driver kit includes the driver that supports the iSCSI protocol. The iSCSI driver kit consists of the following:

- A zipped tar file that includes the driver binary RPMs for a specific driver version, and for all of the supported Linux distribution kernels.

  Note: The driver binary RPM packages only support officially-released Linux distribution kernels, and do not support subsequent errata kernels or pre-release distribution kernels.

- An installation script, elx_iscsi_install.sh, which installs by default the iSCSI driver binary RPM that corresponds to the target system's architecture and kernel memory variant.
- A README file that provides a description of the kit structure, its contents, and distribution support scope.

Once the RPM is installed, the elx_net_install script creates a new ramdisk for the currently running kernel so that the iSCSI driver is loaded when the kernel is initialized during system startup.

Note: The OneCommand Manager application must be installed separately from the driver. Refer to the “Installing the OneCommand Manager Application” on page 9 for more information.

To install the iSCSI driver for Linux:

1. Install a supported Emulex adapter in the system. Refer to the adapter's installation manual for specific hardware installation instructions.
2. Remove any previously installed iSCSI driver kits and/or Application Helper Modules that were installed from the Emulex CD or downloaded from the Emulex Web site, (i.e. not part of a distribution’s kernel) before proceeding. Refer to “Uninstalling the Source-based RPM FC and FCoE Driver Kit” on page 9 for more information.
3. Download the driver kit from the Emulex Web site or copy it to the system from the installation CD.

4. Log on as 'root' to a terminal, and unpack the tarball with the following command:

   ```
   tar xzf elx-be2iscsi-dd-<driver version>.tar.gz
   ```

5. Change to the directory that is extracted:

   ```
   cd elx-be2iscsi-dd-<driver version>/
   ```

6. Execute the 'elx_iscsi_install' script with no options to install the new driver kit. Type:

   ```
   ./elx_iscsi_install.sh
   ```

   Once the elx_iscsi_install' script has completed successfully, the Emulex iSCSI driver is loaded and devices that are properly connected to the system are accessible. Reboot the system now to enable the newly added driver options in the ramdisk. You can also reboot the system later if you want.

Uninstalling the iSCSI Driver Kit

Note: You must run the uninstall script that shipped with the version of the driver kit you want to remove.

To uninstall the iSCSI driver:

1. Log on as 'root'.
2. If possible, exit all applications that use iSCSI-attached drives, then unmount the drives. If you cannot exit all applications that use iSCSI-attached drives, the uninstall will work properly, but you must reboot after the uninstallation is complete.
3. Execute the ‘elx_iscsi_install.sh’ script. with the '--uninstall' option. Type:

   ```
   ./elx_iscsi_install.sh --uninstall
   ```

Booting From a Non-Zero LUN Attached to an Emulex Adapter

This section describes how to configure SLES 11 SP1 to boot from an FC-attached disk device other than /dev/sda. This example uses /dev/sdb.

To boot from a non-zero LUN attached to an LPFC adapter:

1. Configure the Emulex adapter bootBIOS to boot from the desired LUN.
2. Start the standard SLES 11 SP1 installation.
3. At the Installation Settings screen, after configuring the desired partitions, select the **Expert** tab.
4. Select **Booting** to change the bootloader configuration.
5. The Boot Loader Settings window appears. Select the **Boot Loader Installation** tab.
6. In the section labeled Boot Loader Location, select **Custom Boot Partition**, then select the **root partition** (or **boot partition** if you configured one) from the drop-down box.
7. Click the **Boot Loader Options** button. The Boot Loader Options window appears. Select the **Write generic Boot Code to MBR** checkbox.
8. Click **OK**.
9. In the Boot Loader Settings window, Click **Finish**.
10. Proceed with the installation.
11. During the first boot after the installation, use the GRUB command line to change all hd1 references to hd0, then continue the boot process.

12. Edit the GRUB configuration in /boot/grub/menu.lst to change all hd1 references to hd0.

Installing the OneCommand Manager Application

The OneCommand Manager application is a powerful, centralized adapter management suite, providing discovery, reporting and management of local and remote adapters from a single console anywhere in the SAN and across platforms. Both a graphical user interface (GUI) and command line interface (CLI) are provided. This remote configuration capability can be provided by either Fibre Channel (FC) access via host systems on the same FC Storage Area Network (SAN) or by Transmission Control Protocol/ Internet Protocol (TCP/IP) access from IP addresses of remote machines.

Refer to the *OneCommand Manager Application Version 5.1 User Manual*, which is available on the Emulex Web site, for instructions on installing and using the OneCommand Manager application.

Uninstalling the Source-based RPM FC and FCoE Driver Kit

Note: Driver parameter changes made using the OneCommand Manager application or /etc/modprobe.conf persist if the driver is uninstalled. To return to the default settings, you must modify the settings in /etc/modprobe.conf.

Note: You must run the uninstall script that shipped with the version of the driver kit you want to remove.

This section describes how to uninstall a previous version of the Emulex 8.x driver for Linux. The uninstall procedure automatically restores the archived FC and FCoE driver.

To uninstall the source-based RPM FC and FCoE driver:

1. Log on as 'root'.
2. If possible, exit all applications that use Fibre Channel-attached drives, then unmount the drives. If you cannot exit all applications that use Fibre Channel-attached drives, the uninstall will work properly, but you must reboot after the uninstallation is complete.
3. Stop the OneCommand Manager application. Type:

   ```
   cd /usr/sbin/hbanyware
   ./stop_ocmanager
   ```
4. Uninstall the Applications Kit. Refer to the *OneCommand Manager Application Version 5.1 User Manual* on the Emulex Web site for instructions.
5. Copy the lpfc-install script to the temporary directory. For example:

   ```
   cp /usr/src/lpfc/lpfc-install /tmp
   ```
6. Execute the LPFC-install script. with the '--uninstall' option. Type:

   ```
   /tmp/lpfc-install --uninstall
   ```

Configuration

You can configure the FC and FCoE driver by:

- Setting module parameters using modprobe and /etc/modprobe.conf.
- Using the sysfs interface (for parameters that can be changed after loading the FC and FCoE driver).
- Using the OneCommand Manager application. See the *OneCommand Manager Application Version 5.1 User Manual* for more information.

Note: FC and FCoE driver parameter changes made using modprobe.conf or the OneCommand Manager application persist if the FC and FCoE driver is uninstalled. To return to the default settings, you must modify the settings in modprobe.conf.

FC and FCoE Driver Configuration Methods Using modprobe and /etc/modprobe.conf

The following sections describe how to set FC and FCoE driver parameters using the modprobe command and by manually editing /etc/modprobe.conf.

Note: Emulex recommends using the OneCommand Manager application or the OneCommand Manager Application CLI to change parameters. See the *OneCommand Manager Application Version 5.1 User Manual* for more information.

Temporary Configuration Method

When you manually load the FC and FCoE driver as a module using the modprobe command and change one or more driver parameter values, it is a temporary configuration. These changes are considered temporary because they are valid for the current session only or until the FC and FCoE driver is unloaded again. Modprobe uses the modprobe.conf file, but parameters passed to it using the command line override parameters in the modprobe.conf file.

Values can be expressed in hexadecimal or decimal notation.

Example of Temporary Configuration

You want to temporarily set lun_queue_depth to 20 (default is 30) for all host bus adapters in your system. Load the FC and FCoE driver with the following command:

```
modprobe lpfc lpfc_lun_queue_depth=20
```

Persistent Configuration Method

To make the FC and FCoE driver parameters persistent across module loads and reboots, modify the /etc/modprobe.conf file. If driver parameters are modified in /etc/modprobe.conf, the FC and FCoE driver must be reloaded for the parameters to take effect. Also a new ramdisk image is required if you want the changes to take effect in the next boot. See “Creating a New Ramdisk Image” on page 12 to learn how.

The FC and FCoE driver parameters are specified in /etc/modprobe.conf via the “options” command. For example the following sets the verbose flag.

```
options lpfc lpfc_log_verbose=0xffff
```

If the same option is specified in both the /etc/modprobe.conf and on the modprobe command line, the option setting in the command line takes precedence.

Temporary FC and FCoE Driver Configuration by Read/Write to sysfs

Sysfs is a virtual filesystem that exposes the structure of the system. It also includes interfaces to driver parameters through which the FC and FCoE driver parameters can be viewed and modified. Since these interfaces are available only after driver load, only those parameters that can be modified dynamically can be changed. However, all FC and FCoE driver parameters can be read through sysfs.

Note: Sysfs changes only exist during driver load and are lost when the FC and FCoE driver is unloaded or the system is rebooted.

The sysfs filesystem is mounted and available as /sys. You must first identify the scsi_host which represents the adapter for which you want to modify the FC and FCoE driver parameters. All scsi_hosts bound to the FC and FCoE driver can be viewed with the following command:

```
# ls -d /sys/bus/pci/drivers/lpfc/*/host*
```

Assuming you are interested in adapter scsi_host 7, you can list the FC and FCoE driver parameters for this particular adapter as:

```
#ls -l /sys/class/scsi_host/host7/lpfc*
```

An example output is as follows:

```
-r--r--r--  1 root root 4096 Feb 28 17:03 /sys/class/scsi_host/host7/lpfc_ack0
-r--r--r--  1 root root 4096 Feb 28 17:03 /sys/class/scsi_host/host7/lpfc_fcp_class
-rw-r--r--  1 root root 4096 Feb 28 17:03 /sys/class/scsi_host/host7/lpfc_fdmi_on
-r--r--r--  1 root root 4096 Feb 28 17:03 /sys/class/scsi_host/host7/lpfc_link_speed
-rw-r--r--  1 root root 4096 Feb 28 15:34 /sys/class/scsi_host/host7/lpfc_log_verbose
-r--r--r--  1 root root 4096 Feb 28 17:03 /sys/class/scsi_host/host7/lpfc_lun_queue_depth
-rw-r--r--  1 root root 4096 Feb 28 17:03 /sys/class/scsi_host/host7/lpfc_max_luns
-rw-r--r--  1 root root 4096 Feb 28 17:03 /sys/class/scsi_host/host7/lpfc_nodev_tmo
-rw-r--r--  1 root root 4096 Feb 28 17:03 /sys/class/scsi_host/host7/lpfc_scan_down
-r--r--r--  1 root root 4096 Feb 28 17:03 /sys/class/scsi_host/host7/lpfc_topology
-rw-r--r--  1 root root 4096 Feb 28 17:03 /sys/class/scsi_host/host7/lpfc_use_adisc
```

Notice that the FC and FCoE driver parameters are available as files. Reading a file displays the current value of a driver parameter. If the permissions allow it, you can write a value to the file and it will take effect immediately.

For example:

```
[root@emulex]# cat /sys/class/scsi_host/host7/lpfc_log_verbose
0
```

Notice that the current value of lpfc_log_verbose is zero. To set it to 0xffff:

```
[root@emulex]# echo 0xffff > /sys/class/scsi_host/host7/
lpfc_log_verbose
[root@emulex]# cat /sys/class/scsi_host/host7/lpfc_log_verbose
0xffff
```

Creating a New Ramdisk Image

The lpfc-install script creates a ramdisk containing the FC and FCoE driver for the currently running kernel.

Note: You must perform this step whenever the LPFC options in /etc/modprobe.conf are changed and you want the change to take effect on the next reboot.

For Installed FC and FCoE Driver Kits

To create a new initial ramdisk image:

1. su to 'root'.
2. Type:
   ```
   cd /usr/src/lpfc
   ```
3. Execute the lpfc-install script using the '--createramdisk' option. Type:
   ```
   ./lpfc-install --createramdisk
   ```

For Distribution In-Box FC and FCoE Drivers

To create a new initial ramdisk image:

- For SLES11 SP1 distributions type:
  ```
  # mkinitrd -k vmlinuz -i initrd
  ```
- For RHEL6 distributions type:
  ```
  # mkinitrd -f /boot/initrd-<kernel-version>.img <kernel-version>
  ```

Dynamically Adding LUNs and Targets

The FC and FCoE driver enables you to dynamically add LUNs and targets without unloading or reloading the lpfc module and without resetting the adapter.

To rescan an adapter's targets with sysfs given the adapter's host number (in this example 3), type:

```
echo "- - -" > /sys/class/scsi_host/host3/scan
```

To limit the rescan to a particular target, given the adapter's host number (in this example 3) and the target number (in this example 2), type:

```
echo "- 2 -" > /sys/class/scsi_host/host3/scan
```

You can also use the Emulex lun_scan script in /usr/sbin/lpfc.

FC and FCoE Driver Parameters Reference Table

The FC and FCoE driver parameters determine some aspects of the driver's behavior. The following tables list the FC and FCoE driver parameters. Some FC and FCoE driver parameters can be modified and take effect only on a driver load while others can be modified dynamically and take effect immediately. The tables also list the default, minimum and maximum values for these parameters.

Table 2: LPFC Static Parameters (Requires a driver reload to change)

Variable	Default	Min.	Max.	Comments	Visible using sysfs
lpfc_ack0	0	0=Off	1=On	Uses ACK0 for class 2.	Yes
lpfc_discovery_threads	32	1	64	Specifies the maximum number of ELS commands that can be outstanding for a discovery. **Note:** The discovery_threads parameter defaults to a value of 64 for private loop topologies regardless of the configured value. If there are multiple ports configured on the host the value of 64 is only used for those ports that are connected in a private loop topology. The configured value is used for all other ports.	No
lpfc_enable_da_id	0	0 = Disabled (default) 1 = enable – a DA_ID CT command is sent to the fabric when logging out.		This parameter controls whether the FC and FCoE driver will issue a DA_ID CT command to the fabric when VPorts logout of the fabric.	Yes
lpfc_enable_hba_ heartbeat	1	0 = heartbeat disabled 1 = heartbeat enabled		Controls the adapter heartbeat logic in the FC and FCoE driver. If the heartbeat is enabled and the heartbeat logic detects that the adapter is nonfunctional, the FC and FCoE driver will shutdown the adapter.	Yes

Table 2: LPFC Static Parameters (Requires a driver reload to change) (Continued)

Variable	Default	Min.	Max.	Comments	Visible using sysfs
lpfc_enable_hba_reset	1	0 = hba reset disabled 1 = hba reset enabled		Controls whether hba_resets will be allowed by the FC and FCoE driver to pass to the adapter. This is used as a debugging tool.	Yes
lpfc_enable_npiv	0	0	1	This parameter controls the FC and FCoE driver's ability to use NPIV to create virtual ports. It defaults to off (0) which prevents the FC and FCoE driver from creating any virtual ports. When enabled (set to 1) it enables you to create and delete virtual ports (if supported by the fabric).	Yes
lpfc_fcp_class	3	2	3	The FC class for FCP data transmission.	Yes
pfc_fcp_eq_count	4	1	8	Sets the number of fast-path FCP event queues, if available. Only applicable for OneConnect UCNAs.	Yes
lpfc_fcp_imax	10000	636	651042	Sets the maximum number of fast-path FCP interrupts per second. Only applicable for OneConnect UCNAs.	Yes
lpfc_fcp_wq_count	4	1	32	Sets the number of fast-path FCP work queues, if available.Only applicable for OneConnect UCNAs.	Yes
lpfc_hba_queue_depth	8192	32	8192	The maximum number of FCP commands that can queue to an Emulex adapter.	Yes
lpfc_lun_queue_depth	30	1	128	The default maximum commands sent to a single logical unit (disk).	Yes
lpfc_restrict_login	1	0=Off	1=On	Restricts virtual ports login to remote initiators.	No
lpfc_scan_down	1	0=Off	1=On	Selects method for scanning ALPA to assign a SCSI ID.	Yes

Table 2: LPFC Static Parameters (Requires a driver reload to change) (Continued)

Variable	Default	Min.	Max.	Comments	Visible using sysfs
lpfc_sg_seg_cnt	64 (50 for SLI-4 CNA)	64 (50 for SLI-4 CNA)	4096	Controls the max scatter gather segment count passed to the FC and FCoE driver. **Note:** This variable is per SCSI command. On OneConnect UCNAs, these values are restricted by the FC and FCoE driver to specific values due to restrictions imposed by the hardware. The possible values are 50, 114, 242, and 498.	Yes. Displayed as sg_tablesize
lpfc_sli_mode	0	0 = auto (default) 2 = SLI 2 mode 3 = SLI 3 mode		This parameter allows you to force the SLI mode requested by the adapter driver. This parameter has no effect on OneConnect UCNAs.	No
lpfc_max_luns	255	0	65535	Specifies the maximum number of LUN IDs per target. A value of 19 means LUN IDs from 0 to 19 are valid. The SCSI layer scans each target until it reaches the specified LUN ID.	Yes
lpfc_max_scsicmpl_time	0	0	6000	Uses command completion time to control queue depth.	Yes
lpfc_multi_ring_rctl	4	1	255	Identifies RCTL for additional ring configuration. **Note:** Only used when multi_ring_support is enabled.	Yes
lpfc_multi_ring_support	1	1	2	Determines the number of primary SLI rings over which to spread IOCB entries.	Yes
lpfc_multi_ring_type	5	1	255	Identifies TYPE for additional ring configuration. **Note:** Only used when multi_ring_support is enabled.	Yes

Table 2: LPFC Static Parameters (Requires a driver reload to change) (Continued)

Variable	Default	Min.	Max.	Comments	Visible using sysfs
lpfc_use_msi	0	0 = MSI disabled 1 = MSI enabled 2 = MSI-X enabled		Controls whether the driver uses Message Signaled Interrupts.	Yes

All LPFC dynamic parameters are read/write using sysfs.

Table 3: LPFC Dynamic Parameters (Do not require a driver reload to change)

Variable	Default	Min	Max	Comments
lpfc_cr_count	1	1	255	This parameter determines the values for I/O coalescing for cr_count outstanding commands. Not applicable for OneConnect UCNAs.
lpfc_cr_delay	0	0	63	This parameter determines the values for I/O coalescing for cr_delay (msec) outstanding commands. Not applicable for OneConnect UCNAs.
lpfc_devloss_tmo	30	0	255	Seconds to hold I/O error if device disappears.
lpfc_fdmi_on	0	0	2	False (0) if disabled. (1) or (2) if enabled depending on type of support needed.
lpfc_link_speed	0	0=auto select 1=1 Gb/s 2=2 Gb/s 4=4 Gb/s 8=8 Gb/s		Sets link speed. **Note:** This variable does not effect FCoE 10 Gb/s capable adapters.
lpfc_log_verbose	0x0	0x0	0xffff	(bit mask) Extra activity logging.
lpfc_nodev_tmo (depreciated)	30	1	255	Seconds to hold I/O error if device disappears. This parameter will not work if you altered lpfc_devloss_tmo. **Note:** This is a deprecated field and lpfc_devloss_tmo should be used instead.
lpfc_pci_max_read	2048	512, 1024, 2048, 4096		Maximum DMA read byte count.
lpfc_poll	0	1= poll with interrupts enabled 3 = poll and disable FCP ring interrupts		Sets FCP ring polling mode control.
lpfc_poll_tmo	10	1	255	Milliseconds the driver waits between polling FCP ring interrupts.

Table 3: LPFC Dynamic Parameters (Do not require a driver reload to change) (Continued)

Variable	Default	Min	Max	Comments
lpfc_topology	0	0x0=loop then P2P 0x2=P2P only 0x4=loop only 0x6=P2P then loop		FC link topology (defaults to loop, if it fails attempts point-to-point mode). Not applicable for OneConnect UCNAs.
lpfc_use_adisc	0	0=Off	1=On	Sends ADISC instead of PLOGI for device discovery or RSCN.

Using udev for Persistent Naming

SLES 11 is configured by default with udev to provide persistent names for hard disks, including FC attached disks.

Using udev to Discover Logical to Physical Mappings for sd Devices

Persistent names for sd devices are provided in the /dev/disk/by-id directory.

To find the persistent udev name for the disk which is currently sdc, type:

```
# cd /dev/disk/by-id
# ls -l | grep sdc
```

The sample output is shown below:

```
lrwxrwxrwx 1 root root  9 2006-08-01 19:08 scsi-32000000c5005d6e6 -> ../../sdc
```

In the above example, the disk has no partitions. If the disk had two partitions, the output would look like the following:

```
lrwxrwxrwx 1 root root  9 2006-08-01 19:08 scsi-32000000c5005d6e6 -> ../../sdc
lrwxrwxrwx 1 root root 10 2006-08-01 19:08 scsi-32000000c5005d6e6-part1 -> ../../sdc1
lrwxrwxrwx 1 root root 10 2006-08-01 19:08 scsi-32000000c5005d6e6-part2 -> ../../sdc2
```

Configuring the System to Boot From SAN Using Persistent Names

To use a persistent name for a boot device (SLES 11 SP1):

1. In /boot/grub/menu.lst, find the kernel line for the default boot. For example:

   ```
   kernel /boot/vmlinuz root=/dev/sda2 vga=0x314
   ```
2. Find the persistent name for the root partition (following “root=” on the kernel line) by using the instructions in “Using udev to Discover Logical to Physical Mappings for sd Devices” on page 17.
3. In the same file, /boot/grub/menu.lst, replace the text after “root=” with the partition's persistent name. For example:

   ```
   kernel /boot/vmlinuz root=/dev/disk/by-id/scsi-32000000c5005d6e6-part2 vga=0x314
   ```
4. Change any mounts listed in /etc/fstab which refer to this root partition by either it's /dev/sd name or a file system LABEL to use the persistent name as well.

To use a persistent name for a boot device (RHEL 6):

1. In /boot/grub/grub.conf, find the kernel line for the default boot. For example:

   ```
   kernel /boot/vmlinuz -<kernel version> ro root=/dev/sda2
   ```

2. Find the persistent name for the root partition (following "root=" on the kernel line) by using the instructions in “Using udev to Discover Logical to Physical Mappings for sd Devices” on page 17.
3. In the same file, /boot/grub/menu.lst, replace the text after "root=" with the partition's persistent name. For example:

```
   kernel /boot/vmlinuz -<kernel version> ro root=/dev/disk/by-id/scsi-
32000000c5005d6e6-part2
```

4. Change any mounts listed in /etc/fstab which refer to this root partition by either it's /dev/sd name or a file system LABEL to use the persistent name as well.

Using udev with st Devices

The udev rules for tape devices are the same for disk devices. There must be a unique ID that persists across initiator reboots and persists regardless of discovery order.

Another thing to consider is whether or not the tape device is one of many SCSI tape devices residing behind an FC controller, or if it is an FC-Tape device. If it an FC-Tape device, then the WWPN is unique and can be used to create the persistent name. In fact, the scsi_id program should return this as the unique identifier with a single digit prefix.

If the FC controller has multiple SCSI tape devices behind it, the WWPN is not unique and the persistent name must use multiple information elements to build the unique ID.

Below are examples of each scenario. The first example is that of an FC-Tape device. This example uses SCSI generic (sg) rather than the SCSI tape driver.

```
[root@localhost ~]# scsi_id -g -s /sys/class/scsi_generic/sg0
   350060b000029b592
```

The value returned has a leading prefix of 3. This value is the NAA type and what follows is the controller's WWPN.

Below is an example of the same tape device and a scsi_id call. The response is the same.

```
[root@localhost ~]# scsi_id -g -s /sys/class/scsi_tape/nst0
   350060b000029b592
```

In both examples, -g was needed because the vendor and model for this tape device were not in /etc/scsi_id.config.

Below is another example for a different FC-Tape Vendor. Notice that the answer is similar with respect to the leading digit and the WWPN.

```
[root@localhost ~]# /sbin/scsi_id -g -s sys/class/scsi_tape/nst0
35005076300015101
```

Below is an example of a FC-SCSI Tape device. Notice that when the Emulex driver loads, the SCSI midlayer discovers the SCSI tape devices as follows:

```
scsi scan: INQUIRY to host 14 channel 0 id 0 lun 0
scsi: unknown device type 12
Vendor: ADIC      Model: SNC 4000          Rev: 42d4
Type:   RAID                               ANSI SCSI revision: 03
Attached scsi generic sg5 at scsi14, channel 0, id 0, lun 0,  type 12
scsi scan: INQUIRY to host 14 channel 0 id 0 lun 1
Vendor: ADIC      Model: Scalar 24         Rev: 227A
Type:   Medium Changer                     ANSI SCSI revision: 02
Attached scsi generic sg6 at scsi14, channel 0, id 0, lun 1,type 8
scsi scan: INQUIRY to host 14 channel 0 id 0 lun 2
Vendor: IBM       Model: ULTRIUM-TD2       Rev: 38D0
Type:   Sequential-Access                  ANSI SCSI revision: 03
Attached scsi tape st0 at scsi14, channel 0, id 0, lun 2
st0: try direct i/o: yes (alignment 512 B), max page reachable by HBA
4503599627370495
```

```
Attached scsi generic sg7 at scsi14, channel 0, id 0, lun 2,  type 1
scsi scan: INQUIRY to host 14 channel 0 id 0 lun 3
Vendor: IBM       Model: ULTRIUM-TD2       Rev: 38D0
Type:   Sequential-Access                  ANSI SCSI revision: 03
Attached scsi tape st1 at scsi14, channel 0, id 0, lun 3
st1: try direct i/o: yes (alignment 512 B), max page reachable by HBA
4503599627370495
Attached scsi generic sg8 at scsi14, channel 0, id 0, lun 3,  type 1
```

This log output shows a controller at LUN 0, the medium changer at LUN 1 and two SCSI tape devices at LUNs 2 and 3. The example below is what the scsi_id call returns:

```
[root@localhost ~]# scsi_id -g -s /sys/class/scsi_tape/nst0
1IBM     ULTRIUM-TD2     1110133831
[[root@localhost ~]# scsi_id -g -s /sys/class/scsi_tape/nst1
1IBM     ULTRIUM-TD2     1110133994
```

Notice that the unique ID is actually comprised of three value with space delimiters. A udev rule must have a unique ID for the device, meaning all three parts of this returned string are required. To do this, use the following command.

```
[root@localhost ~]# scsi_id -u -g -s /sys/class/scsi_tape/nst0
1IBM_____ULTRIUM-TD2_____1110133831
[root@localhost ~]# scsi_id -u -g -s /sys/class/scsi_tape/nst1
1IBM_____ULTRIUM-TD2_____1110133994
```

Creating the udev persistent name for SCSI tape uses the same process as SCSI disk once the SCSI ID call needed to extract a unique ID is known.

Below is the rule for the FC-Tape device:

```
BUS="scsi", SYSFS{vendor}="HP", SYSFS{model}="ULTRIUM 3-SCSI",
PROGRAM="/sbin/scsi_id -p 0x83 -u -g -s /sys/class/scsi_tape/
nst%n",RESULT="350060b000029b592", SYMLINK="fc_lun_st%n"
```

The rule for the FC-SCSI tape device follows:

```
BUS="scsi", SYSFS{vendor}="IBM", SYSFS{model}="ULTRIUM-TD2",
PROGRAM="/sbin/scsi_id -p 0x83 -u -g -s /sys/class/scsi_tape/
nst%n",RESULT="1IBM_____ULTRIUM-TD2_____1110133831",
SYMLINK="fc_lun_st%n"
BUS="scsi", RESULT="1IBM_____ULTRIUM-TD2_____1110133994",
SYMLINK="fc_lun_st%n"
```

Create a new file named /etc/udev/rules.d/45-local.rules and put the appropriate rule in it. Then run udevtrigger to reload the udev rules.

And finally, here is the output of the rule:

```
[root@localhost ~]# udevtrigger
[root@localhost ~]# ls -al /dev/fc*
lrwxrwxrwx  1 root root 3 Apr  7 15:03 fc_lun_st0 -> st0
lrwxrwxrwx  1 root root 3 Apr  7 15:03 fc_lun_st1 -> st1
```

Further Information About Persistent Names

Refer to the following references for more information on persistent naming:

http://www.reactivated.net/udevrules.php by Daniel Drake (dsd)

http://kernel.org/pub/linux/utils/kernel/hotplug/udev_vs_devfs by Greg Kroah-Hartman

http://linux.dell.com/devlabel/devlabel.htm

Working with Virtual Ports (VPorts)

Creating, Deleting and Displaying VPorts

VPorts are created through sysfs entries that are presented in the physical port's sysfs directory. The vport_create and vport_delete sysfs entries are discussed in the sysfs section, but there are also three scripts for creating, deleting and displaying VPorts. The scripts reside in the /usr/sbin/lpfc directory and are part of the OneCommand Manager Applications kit.

When NPIV is enabled and VPorts are configured it may take longer for the adapter to finish discovery in some cases due to the fact that each virtual port must perform discovery independently. As more VPorts are configured the amount of time that the driver and adapter take to finish discovery of remote ports on the SAN will increase. To compensate for this extended amount of time taken in discovery it is recommended that the user set the lpfc_devloss_tmo parameter to 60 when npiv is enabled.

Note: Ensure you are using the latest recommended firmware for VPort functionality. Check the Emulex Web site for the latest firmware.

Note: Loop devices and NPIV are not supported on the same port simultaneously. If you are running a loop topology and you create a VPort, the VPort's link state will be off line.

Note: You can only create virtual ports on 4 Gb/s, 8 Gb/s and 10 Gb/s adapters. You cannot create virtual ports on 1 Gb/s and 2 Gb/s adapters.

The mkvport.sh Script

You can use the mkvport script to create VPorts. To see the usage information, run the script with no parameters specified. The mkvport.sh script uses the following syntax:

```
./mkvport.sh <Physical Port's Host number> <Port Name> <Node Name>
```

For example:

```
> ./mkvport.sh host7 10000000c94ac63a 20010000c94ac63a
```

would create a VPort with port name of 10000000c94ac63a and a node name of 20010000c94ac63a on the physical port with scsi_host name “host7”. This script will fail if the VPort is not created.

Note: You must supply the physical port’s host number, WWPN and WWNN when using the mkvport.sh script.

Note: It is possible for a VPort to be created successfully, but be in “failed” state. For example, loop devices and NPIV are not supported on the same port simultaneously. If you are running a loop topology and you create a VPort, the VPort's link state will be off line.

The rmvport.sh Script

You can use the rmvport script to delete VPorts. To see the usage information, run the script with no parameters specified. The rmvport.sh script uses the following syntax:

```
./rmvport.sh <Virtual Port's Host number>
```

Or

```
./rmvport.sh <Port Name> <Node Name>
```

For example

```
> ./rmvport.sh 10000000c94ac63a 20010000c94ac63a
```

would delete the VPort with port name of 10000000c94ac63a and node name of 20010000c94ac63a. This script will fail if the VPort is not deleted and may take up to 30 seconds to complete.

Note: You must un-map, un-mount, and flush I/O to VPort connected devices before deleting the VPort.

The lsvport.sh Script

You can use the lsvport script to list the VPorts and physical ports that are present on the system. Run the script with no parameters to display port information.

For example:

```
[root@curly scripts]# ./lsvport.sh
lpfc0: host6 10000000c93a5b5e:20000000c93a5b5e LP10000 NPIV Not Supported
lpfc1: host7 10000000c93a5b5d:20000000c93a5b5d LP10000 NPIV Not Supported
lpfc2: host8 10000000c93cc8dd:20000000c93cc8dd LPe12000 NPIV Physical
  lpfc4: host10 10000000c94ac63a:20010000c94ac63a  NPIV Virtual (VPI 1)
lpfc3: host9 10000000c93cc8dc:20000000c93cc8dc LPe12000 NPIV Physical
[root@curly scripts]#
```

For LPFC0 and LPFC1, “NPIV Not Supported” means that this adapter/firmware combination does not support the creation of VPorts.

For LPFC2, “NPIV Physical” refers to a physical port of this adapter.

For LPFC4, “NPIV Virtual” refers to a VPort of this adapter.

The VPort Sysfs Tree

When a VPort is created, three new directories are created in the class tree:

```
/sys/class/scsi_host/hostY/
/sys/class/fc_host/hostY/
/sys/class/fc_vports/vport-X:0-Z/-
```

Creating a new VPort also creates a new sysfs directory in the bus and devices tree:

```
/sys/bus/pci/drivers/lpfc/0000:A:B:C/hostX/vport-X:0-Z/hostY
/sys/devices/pci0000:A/0000:A:B:C/hostX/vport-X:0-Z/hostY
```

In both directories there is a hostY directory that contains the remote ports that this new host can access:

```
/sys/bus/pci/drivers/lpfc/0000:A:B:C/hostX/vport-X:0-Z/hostY
/sys/bus/pci/drivers/lpfc/0000:A:B:C/hostX/vport-X:0-Z/hostY/rport-Y:0-0
/sys/bus/pci/drivers/lpfc/0000:A:B:C/hostX/vport-X:0-Z/hostY/rport-Y:0-1
/sys/bus/pci/drivers/lpfc/0000:A:B:C/hostX/vport-X:0-Z/hostY/rport-Y:0-2
```

“Y” indicates the new host value for the virtual port that was created.

“X” indicates the host value for the parent fc_host that this virtual port was created from.

“Z” indicates the instance of virtual port created from the parent fc_host. A, B, and C indicate the PCI hierarchy for each physical LPFC port.

In other words, hostY is the new host created for the new virtual port. vport-X:0-Z uniquely identifies the VPort and indicates the parent host structure (XXX) that this virtual port was created by.

For example, when we create a VPort from host5 we get a new scsi_host, fc_host, fc_vport, and a new entry under the bus tree as well.

```
[root@doc ~]# ls /sys/class/scsi_host/
host0  host1  host4  host5  host6
[root@doc ~]# ls /sys/class/fc_host/
host4  host5  host6
[root@doc ~]# ls /sys/class/fc_vports/
vport-5:0-0
```

FC and FCoE Driver Version 8.3.5.x sysfs Structure

In the 8.3.5.x FC and FCoE driver, the transport creates an fc_vport directory that you can use to monitor VPorts. This directory is populated entirely of VPorts and has links from each to the fc_host associated with that VPort.

```
[root@doc ~]# ls /sys/class/fc_vports/
vport-5:0-0
[root@doc ~]# ls -d /sys/bus/pci/drivers/lpfc/*/host*/*/host*
/sys/bus/pci/drivers/lpfc/0000:03:06.1/host5/vport-5:0-0/host6
[root@doc ~]# ls /sys/devices/pci*/*/host5/vport-5*/host6
power  rport-6:0-0  rport-6:0-1  rport-6:0-2  uevent
[root@doc ~]# ls /sys/devices/pci*/*/host5/vport-5*/host6/rport-*
/sys/devices/pci00:03/00:03:06.1/host5/vport-5:0-0/host6/rport-6:0-0:
power  uevent

/sys/devices/pci00:03/00:03:06.1/host5/vport-5:0-0/host6/rport-6:0-1:
power  uevent

/sys/devices/pci00:03/00:03:06.1/host5/vport-5:0-0/host6/rport-6:0-2:
power  target6:0:0  uevent
```

The new host for the virtual port is host6. It appears in the usual directories and now there is a new entry in the fc_vports directory for the VPort that indicates that the VPort was created from host5 and is the first (0) VPort to be created on that fc_host. There is also a new directory in the bus tree to indicate that host6 was created under vport-5:0-0 that was created from host5.

VPort sysfs Entries

The following table describes VPort sysfs entries.

Note: VPort sysfs entries in Table 5 are only present if the driver was loaded with lpfc_enable_npiv enabled.

Table 4: VPort sysfs Entries

VPort sysfs Entries	Type	Range/ Input	Location and Description
npiv_vports_inuse	read-only	integers	`/sys/class/fc_host/hostX/npiv_vports_inuse` This entry displays the number of VPorts that were created on this fc_host. This sysfs entry will only exist if the vport_create and vport_delete sysfs entries exist. If an fc_host does not support NPIV then this sysfs entry may not exist. **Note:** Use this sysfs entry along with max_npiv_vports to determine whether the maximum number of VPorts have been created on this fc_host.
max_npiv_vports	read-only	integers	`/sys/class/fc_host/hostX/max_npiv_vports` This entry displays the maximum number of VPorts that are supported by the fc_hosts underlying hardware. This sysfs entry will only exist if the vport_create and vport_delete sysfs entries exist. If an fc_host does not support NPIV then this sysfs entry may not exist. **Note:** Use this sysfs entry along with npiv_vports_inuse to determine whether the maximum number of VPorts have been created on this fc_host.
vport_create	write-only	WWPN; WWNN	`/sys/class/fc_host/hostX/vport_create` This entry creates a VPort on the physical port that hostX is located on. The new VPort will have present a WWPN and WWNN on the fabric as indicated by the WWPN and WWNN that is input to this sysfs entry. This sysfs entry will return a 0 if the VPort creation was successful. A non-zero value indicates that the VPort failed to be created. If an fc_host does not support NPIV then this sysfs entry may not exist. **Note:** It is possible for the VPort creation to succeed but for the VPort to be in a failed or inoperative state. Use the new sysfs tree created by the new VPort to check the state of the new VPort.

Table 4: VPort sysfs Entries (Continued)

VPort sysfs Entries	Type	Range/ Input	Location and Description
vport_delete	write-only	WWPN; WWNN	`/sys/class/fc_host/hostX/vport_delete` This entry deletes a VPort on the physical port that hostX is located on. The VPort matching the WWPN and WWNN will be immediately deleted. This entry returns a 0 if the VPort deletion was successful. A non-zero value indicates that the VPort failed to be deleted. If an fc_host does not support NPIV then this sysfs entry may not exist. **Note:** This entry will delete the VPort even if there are mounted file systems being accessed through this VPort and/or open files.
node_name	read-only	16 byte hex. value	`/sys/class/fc_host/hostX/node_name/sys/class/fc_vports/vport-X:0-Z/node_name` This entry displays physical or virtual port's node name. This is the value that is assigned by you upon creation and transmitted to the fabric upon fabric login.
port_name	read-only	16 byte hex. value	`/sys/class/fc_host/hostX/port_name/sys/class/fc_vports/vport-X:0-Z/port_name` This entry displays physical or virtual port's port name. This is the value that you assign when you create a VPort. It is transmitted to the fabric upon fabric login.
lpfc_peer_port_login	read/ write	0=Off (default) 1=On	`/sys/class/scsi_host/hostX/lpfc_peer_port_login` This entry sets the port’s behavior when discovering targets in the SAN. The default behavior (0) will only login to nports that are physically located on a different port. The port will still attempt to login to targets on all other ports (including the other port in a dual ported adapter). If this parameter is turned on (1) then the port will attempt to login to all nports, even if they are physically located on the same port. **Note:** This parameter was created to reduce the amount of hardware resources (RPI) that the driver requires. In a configuration where there are many VPorts on one physical port this feature will greatly reduce the number of RPI that the driver utilizes.

VPort Configuration Limits

The following is a list of limits that are supported by the 8.3.5.x driver and configurations that were tested with it. It is highly recommended that you adhere to these limits. Configurations exceeding any one or more of these limits are unsupported. These limits are broken up into two groups. Enforced limits are limits that the driver is able to enforce and will prevent the user from exceeding. Un-enforced limits are limits that the driver cannot enforce and configurations that exceed these limits are unsupported.

Configuration limits:

- All I/O to devices accessed through a VPort must be stopped and all file systems must be unmounted before the VPort is deleted or the driver is unloaded.
- For enterprise class adapters, the maximum number of virtual ports configurable on a physical port is 64. The hardware will allow more than 64 VPorts to be created, but the driver has only been qualified at 64. For mid-range adapters, the maximum number of VPorts configurable on a physical port is 16.
- The maximum number of LUNs supported on each driver port is 256.
- The maximum number of targets supported for each driver port is 255.
- The maximum number of driver ports in one zone is 64. This limit is based on the system's ability to recover from link events within the time constraints of the default timers. The use-cases of NPIV that involve virtual server environment include associating a virtual port with a virtual machine, and placing the virtual machine in its own zone. This will result in one virtual port per zone. In the case of load balanced environments, this can increase typically to two virtual ports per virtual machine, to a practical limit of something far less than 50. In the NPIV cases not related to virtual server environments, zoning will typically be initiator-zoning, again resulting in one virtual port, or a low number of virtual ports in the case of load-balancing, within a given zone. If there are too many virtual ports within a single zone, expected behavior will include devices going lost after link events.
- Minimum lifetime of a virtual port: 60 seconds. There is an un-enforced limit of 60 seconds between the creation of a virtual port and the deletion of the same virtual port. Virtual ports are designed to be an entity that lives for a long time in the system and the creation of VPorts is asynchronous. This means that a virtual port might not be finished with Fibre Channel or SCSI discovery when the command to create a virtual port is finished.
- SMB (3 digit model number) adapters must be zoned so that they can not access adapters with virtual ports configured. SMB adapters have a limited number of resources that make it impossible to operate in the same zone as an adapter that has configured virtual ports.

Network Driver Performance Tuning

Network driver performance tuning improves performance of the network and TCP Offload driver for the Windows Server operating system, Linux Server, and ESX Server. The OneConnect UCNA is an x8, Generation 2 ("Gen 2", or Gen2) PCI-Express device and requires substantial memory bandwidth in a system to support 10 Gb/s data streams.

Improving Performance with PCI-Express Bandwidth

OneConnect UCNA performance may be improved by selecting a more efficient PCI-Express packet payload size. If the system BIOS allows selection of a larger PCI-Express packet size, selecting at least a 512-byte PCIe packet payload size provides the best efficiency for PCIe data transfers.

Improving Performance with TCP Offload

TCP offload helps memory bandwidth significantly by eliminating the data copy of receive packets. This higher memory bandwidth leads to better network performance.

Most computers offer multiple distinct memory channels, or memory interleaves, which may not be enabled by default. Check the manufacturer's documentation and BIOS parameters for details ion enabling optimal memory bandwidth features. Typically, all the DIMM slots must be populated to make use of all the memory channels. As a general rule, more DIMMs provide better performance by allowing a higher degree of memory-access interleaving to occur.

Some servers may allow memory mirroring, where the total memory is divided in half and each location is stored twice. This allows fault recovery if one memory location detects an error, but it greatly reduces the perceived memory bandwidth of the system.

Nearly any desktop or low-end server has enough memory bandwidth for OneConnect UCNAs to support DMA at 20 Gb/s of data (10 Gb/s read, 10 Gb/s write). However, most of the memory demands come from the processor accessing the data for either packet copies in the non-offloaded networking stack or application accesses. All processor memory accesses use the front side bus (FSB). The clock speed of this bus is critical for allowing efficient memory bandwidth.

Note: Systems with a faster Processor Front Side Bus (FSB) clock speed perform better than those with slower FSB clock speeds.

Linux Network Driver

The following section discusses ways to use various OneConnect driver properties and Linux properties to performance tune a system. You can read and set most OneConnect driver settings by using the ethtool utility.

Network Buffer Sizes and TCP Parameters

The optimal size for the network queues and buffers depends on several factors such as protocol, number of streams (connections), request size, and application behavior. The following network configuration settings are a good combination to get best bidirectional transmit and receive performance with six or more TCP connections/UDP streams:

```
echo 4096 87380 4194304 > /proc/sys/net/ipv4/tcp_rmem
echo 4096 16384 4194304 > /proc/sys/net/ipv4/tcp_wmem
echo 64000000 > /proc/sys/net/core/rmem_default
echo 64000000 > /proc/sys/net/core/rmem_max
echo 32000000 > /proc/sys/net/core/wmem_default
echo 32000000 > /proc/sys/net/core/wmem_max
echo 0 > /proc/sys/net/ipv4/tcp_timestamps
echo 0 > /proc/sys/net/ipv4/tcp_sack
ifconfig eth<X> txqueuelen 100
```

The above settings assume ideal conditions such as low latency, zero or close to zero packet loss in the network, enough free memory, and 10 Gb/s path to peer system.

The tcp_rmem and tcp_wmem values above are also the default values in recent updates of the RHEL 6 and SLES 11 SP1 distributions. If the application requires best throughput with very small number of connections (less than four), it may help to increase the tcp_rmem and tcp_wmem to much larger values:

```
echo 4096 87380 16777216 > /proc/sys/net/ipv4/tcp_rmem
echo 4096 65536 16777216 > /proc/sys/net/ipv4/tcp_wmem
```

TCP Segmentation Offload (TSO)

TCP Segmentation Offload (TSO) is enabled by default. In networks with very little loss, TSO improves performance considerably and must remain enabled. The proc variable: /proc/sys/net/ipv4/tcp_tso_win_divisor controls how aggressive the network stack can be in making TSO requests. TSO divisor values in the range 2 to 16 are recommended for a low loss network. The default value of 3 in the RHEL 6 and SLES 11 SP1 distributions seem to be the optimal one for a no loss network.

Smaller divisor values result in larger TSO chunks and better throughput as well as CPU utilization.

However, if the receiver or the network is dropping frames (too many retransmits on transmit side as indicated by netstat -st), it may help to make TSO less aggressive by increasing the divisor value or even turn off TSO. To set the divisor to 8, run:

```
      echo 8 > /proc/sys/net/ipv4/tcp_tso_win_divisor
```

To turn TSO on or off, run the ethtool commands:

```
      ethtool -K <ethX> tso off
      ethtool -K <ethX> tso on
```

where ethX is the name of the Ethernet device you are working on.

Flow Control

Refer to “Flow Control” on page 13 in the Windows Driver section of this document for an understanding of Link layer flow control in 10 Gb/s networks. You can enable and disable the OneConnect UCNA to respond to flow control pause frames from the other side (switch or router) using the following ethtool commands:

```
      ethtool -A <ethN> pause rx on
      ethtool -A <ethN> pause rx off
```

where <ethN> is the number of the Ethernet interface you are working on.

The OneConnect UCNA can be configured to send flow control pause frames using the following ethtool commands:

```
      ethtool -A <ethN> pause tx on
      ethtool -A <ethN> pause tx off
```

where <ethN> is the number of the Ethernet interface you are working on. RX and TX flow control are ON by default.

Refer to the switch/router documentation to determine how link level flow control can be configured on the switch/router to which the OneConnect UCNA port is connected.

RX Frame Coalescing/Large Receive Offload (LRO)

The OneConnect driver consolidates small TCP segments to a large frame before passing to the network stack. This could give considerable boost to TCP receive performance. RX frame coalescing is enabled by default. In some configurations where the end point for the TCP connection to which the packets belong is not in the current server (e.g.: router), RX coalescing should not be enabled. To disable RX coalescing, run the ethtool comand:

```
      ethtool -C <ethN> rx-frames 1
```

where <ethN> is the number of the Ethernet interface on which you are working.

Maximum Transmission Unit (MTU)

The OneConnect driver for Linux supports MTUs between 256 bytes and 9000 bytes. The default MTU is set to 1500. If other elements in the network path support a larger MTU, you can increase the MTU up to 9000 using the ifconfig command. To do this run:

```
      ifconfig <ethN> mtu 9000
```

where <ethN> is the number of the Ethernet interface you are working on.

The largest MTU that does not cause IP fragmentation in the network path gives the best performance. By default, the Linux network stack monitors the lowest path MTU along each open network path (Path MTU Discovery) and adjusts the MSS of established TCP connections to prevent IP fragmentation.

Interrupt Coalescing

On the OneConnect driver, adaptive interrupt coalescing is enabled by default. In light traffic, the interrupt delay is disabled for lower latency. As the number of interrupts/second increases, the delay is increased to the default higher limit of 96 microseconds. You can disable adaptive interrupt coalescing for both RX and TX, by running the ethool command:

```
ethtool -C <ethN> adaptive-rx off
```

where < ethN> is the number of the Ethernet interface you are working on.

When adaptive interrupt coalescing is enabled, the default lower and higher interrupt delay limits of 0 and 96 microseconds can be changed. The coalescing value for rx-usecs and tx-usecs are incremented by 8 ranging from 0 to 96. To do this, run the ethtool commands:

```
ethtool -C <ethN> rx-usecs-high 40
ethtool -C <ethN> rx-usecs-low 8
```

where <ethN> is the number of the Ethernet interface you are working on.

The granularity for delay is 8 microseconds.

If the application requires low/predictive latency, it is recommended that you turn off adaptive interrupt coalescing and set rx-usecs to 0.

CPU Binding Considerations

When using MSI-X, for best performance, the RX and TX interrupts from the OneConnect UCNA must be distributed across all available CPUs. Read /proc/interrupts to see the current distribution of interrupts:

```
# cat /proc/interrupts
          CPU0       CPU1
0:     1556391          0   IO-APIC-edge timer
1:          30       2206   IO-APIC-edge i8042
8:           1          0   IO-APIC-edge rtc
9:           0          0   IO-APIC-level acpi
12:         96          0   IO-APIC-edge i8042
74:       2846       7013   IO-APIC-level libata
82:          0          0   IO-APIC-level uhci_hcd:usb1
98:          0          0   IO-APIC-level uhci_hcd:usb2
114: 104806161          0   PCI-MSI-X eth0-rx
122:  47578488          0   PCI-MSI-X eth0-tx
130:  48014463          0   PCI-MSI-X eth1-rx
138:  17150482          0   PCI-MSI-X eth1-tx
169:         6          1   IO-APIC-level se_ec3210
177:         6          0   IO-APIC-level se_ec3210
NMI:         0          0
LOC:   1555444    1555793
ERR:         1
MIS:         0
```

In the above example, RX and TX interrupts from both port0 (eth0) and port1 (eth1) are directed to CPU0. This could lead to CPU0 becoming very busy, making it a bottleneck. To direct the RX and TX interrupts from port1 to CPU1, write the CPU mask into smp_affinity mask of the corresponding interrupt vector:

```
echo 2 > /proc/irq/130/smp_affinity
echo 2 > /proc/irq/138/smp_affinity
```

If there are more than two cores, it is a good idea to direct all the four OneConnect UCNA interrupts to four different cores. If the CPU has four cores, use the four cores that belong to the same physical CPU for best performance.

In an SMP system, although the scheduler attempts to distribute the load, you can achieve more consistent performance by binding the send/receive processes to the appropriate CPU. To find the appropriate CPU to bind to, first find the current utilization of each CPU using the command top. For example, in a quad core system, if the RX and TX interrupts from port0 and port1 are bound to CPUs 0, 1, 2 and 3 respectively, and while the application/test is running, top shows that CPUs 0 and 2 are extremely busy and CPUs 1 and 3 are relatively idle, it helps to bind the application(s) sending/receiving data to CPUs 1 and 3. This can be done using the taskset command. For example:

```
# taskset -c 1,3 ./netserver
```

Starts the command netserver with affinity to CPUs 1 and 3.

If the application / test is already running, it can be bound to a set of CPUs by specifying the bit mask of the CPUs and the PID of the process. For example, if the PID of the process is 2045:

```
# taskset -p 0xA 2045
```

sets the affinity to CPUs 1 and 3.

Note: For the best send and receive performance, use dual core CPUs with large shared L2 cache.

Use the taskset command in Linux to bind a process to a CPU. For example, to run netserver with affinity to CPU ID 1, run:

```
taskset -c 1 ./netserver
```

MSI-X Interrupts

If the blade server and Linux version support MSI-X, the be2net driver automatically uses MSI-X interrupts. This helps to distribute the RX and TX completion processing load for the two ports across a maximum of four CPU cores and get the best throughput.

SELinux Auditing

Turning off auditing and SELinux can improve CPU utilization and in some cases give better throughput. You can disable auditing by appending audit=0 in the boot command line. You can turn off SELinux by specifying: selinux=0 in the boot command line.

For example, the boot command line:

```
kernel /boot/vmlinux-2.6.18 ro root=/dev/md0 selinux=0 audit=0
```

boots the Linux kernel with selinux and audit features disabled.

You can get better CPU utilization and in some cases better throughput by disabling kernel debug options like CONFIG_DEBUG_SLAB. This requires you to build the kernel image and modules.

Troubleshooting

Introduction

There are several circumstances in which your system may operate in an unexpected manner. The Troubleshooting section explains many of these circumstances and offers one or more workarounds for each situation.

Unusual Situations and their Resolutions

General Situations

Table 5: General Driver Situations

Situation	Resolution
FC link fails to come up.	If an FC link fails to come up, verify that an 8 Gb/s adapter is not attempting to connect to a 1 Gb/s device. Only 2, 4, and 8 Gb/s devices are supported on 8 Gb/s adapters. For LP21000 series adapters, ensure the adapter is not in maintenance mode and that it is not running the manufacturing firmware.
Error states “Authentication is enabled but authentication service is not running.”	If you see this message in /var/log/messages and the adapter is in an “Error” state, the fcauthd daemon probably is not running. To check if fcauthd is running execute /etc/init.d/fcauthd status. To start fcauthd execute /etc/init.d/fcauthd start.
If a SAN configuration has 256 targets mapped by the FC and FCoE driver, any additional added targets do not get a target ID mapping by the driver and cause target discovery to fail. Removing targets or reinitializing the link does not solve the problem.	Unload and reload the driver to reset available target IDs. Ensure that the SAN configuration is correct prior to rebooting the driver. This will clear the driver’s consistent binding table and free target IDs for new target nodes.
In some cases, after loading an OEM supplied combined firmware/OpenBoot image you will not be able to enable BootBIOS from the lputil Boot BIOS Maintenance menu.	1. Download the current OpenBoot only image for your adapter from the Emulex Web site. 2. Load the current OpenBoot only image following steps listed in Updating BootBIOS section of this manual. 3. Run lputil, return to Boot BIOS Maintenance menu. 4. Enable BootBIOS.
rmmod fails to unload FC and FCoE driver module due to ERROR: Module lpfc is in use. This message can appear when you attempt to remove the driver and there is a Logical Volume Group dependent on the driver.	1. Make the Logical Volume Group unavailable. Type: lvchange -a n xxxxxxx where xxxxxx is the Volume Group Name. 2. Stop the OneCommand Manager application. 3. Stop Device Mapper.

Table 5: General Driver Situations (Continued)

Situation	Resolution
rmmod of lpfc driver hangs and module reference count is 0.	Due to a small race condition in the kernel it is possible for an rmmod command to hang. Issue the `rmmod -w` command. If this does not help, reboot the computer.
rmmod fails to unload driver due to Device or resource busy. This message occurs when you attempt to remove the driver without first stopping the OneCommand Manager application or the fcauthd daemon, when the OneCommand Manager application is installed and running or when FC disks connected to a LightPulse adapter are mounted.	Stop the OneCommand Manager application before attempting to unload the driver. The script is located in the /usr/sbin/hbanyware directory. Type: ./stop_ocmanager Unmount any disks connected to the adapter. Unload the driver. Type: rmmod lpfcdfc Type: rmmod lpfc
An lspci will show recent Emulex adapters as “unknown”. This is because of the delay of getting new product ID's into the Linux development cycle.	None at this time.
Slow targets or extended link faults on the storage side may result in storage being marked off-line by the mid-layer and remaining off-line (not recovered) when the link faults are corrected.	This version of the driver should eliminate this problem. However, should you experience off-line device issues, increase the SCSI command timeout to a value greater than or equal to sixty seconds. Emulex also provides a script which addresses this issue (for 2.6 kernels). To access the lun_change_state.sh script, click http://www.emulex.com/support/linux/index.jsp, then click the link to the appropriate driver, and click the Linux tools link.
Under certain conditions of an I/O load, some targets cannot retire an I/O issued by a Linux initiator within the default timeout of 30 seconds given by the scsi midlayer. If the situation is not corrected, the initiator-to-target condition deteriorates into abort/recovery storms leading to I/O failures in the block layer. These types of failures are preceded by a SCSI IO error of hex 6000000.	Emulex provides a script which addresses this issue. To access the set_target_timeout.sh script, click http://www.emulex.com/support/linux/index.jsp, then click the link to the appropriate driver, and click the Linux tools link.
FC and FCoE driver fails to recognize an adapter and logs “unknown IOCB” messages in the system log during driver load. The adapter is running outdated firmware.	Upgrade adapter firmware to minimum supported revision listed in installation guide (or newer).
Loading the FC and FCoE driver on SLES 11 SP1 reports “unsupported module, tainting kernel” in system log.	This message is logged by the kernel whenever a module which is not shipped with the kernel is loaded. This message can be ignored.
System panics when booted with a failed adapter installed.	Remove the failed adapter and reboot.
Unloading the FC and FCoE driver on SLES 11 SP1 causes messages like the following to be logged in the system log: “umount: /dev/disk/bypath/pci-0000:02:04.0-scsi-0:0:1:0: not mounted”	These messages are normal output from the SLES 10 hotplug scripts and can be safely ignored.

Table 5: General Driver Situations (Continued)

Situation	Resolution
Driver Install Fails. The lpfc-install script fails to install the driver.	The install script may fail for the following reasons: • A previous version of the driver is installed. Run the lpfc-install --uninstall script and then try to install the driver. • The current driver is already installed. • Run a supported RHEL or SLES kernel.
“No module lpfc found for kernel” error message. When upgrading the kernel, rpm generates the following error: “No module lpfc found for kernel KERNELVERSION”. **A recently upgraded kernel cannot find the ramdisk.** After upgrading the kernel, the kernel cannot find the ramdisk which halts or panics the system. **The driver is not loaded after a system reboot after upgrading the kernel.**	These three situations may be resolved by upgrading the kernel. There are two ways to install the driver into an upgraded kernel. The method you use depends on whether or not you are upgrading the driver. • Upgrade the kernel using the same version of the driver. • Upgrade the kernel using a new version of the driver. See the Installation section for these procedures.
Driver uninstall fails. The lpfc-install --uninstall script fails with an error.	Try the following solutions: • Uninstall the OneCommand Manager application and SSC software packages. These can be removed by running the ./uninstall script from the OneCommand Manager application installation directory. • Unmount all FC disk drives. • Unload the lpfcdfc and FC and FCoE driver. • Use rpm -e lpfcdriver and -e ocmanager and uninstall the new kits.
lpfc-install script exit code.	The lpfc-install script contains exit codes that can be useful in diagnosing installation problems. See the lpfc-install script for a complete listing of codes and definitions.
The OneCommand Manager **application software package will not install.** An error message states that: “inserv Service Elxlpfc has to be enabled for service ElxDiscSrvinserv: exiting now/sbin/ inserv failed exit code 1."	Reinstall the driver with the lpfc-install script.

Table 5: General Driver Situations (Continued)

Situation	Resolution
The Emulex driver for Linux does not load in ramdisk for a custom built kernel.	Custom built kernels are not supported by Emulex. However, the Emulex install script will attempt to install the driver into a ramdisk that follows the naming scheme used by Red Hat or SLES kernels. • The SLES naming scheme for IA64 ramdisk images is: /boot/efi/efi/suse/initrd. • The SLES naming scheme for ramdisk images on all other architectures is: /boot/initrd. If a custom built kernel has a ramdisk image that does not follow the appropriate naming scheme, the name of the image can be changed using the following procedure: 1. Change the name of the ramdisk image to match the SLES naming scheme. 2. Update any file links to the ramdisk image. 3. Edit the boot loader configuration file: (i.e., /etc/lilo.conf, /etc/yaboot.conf, /boot/grub/grub.conf, /boot/grub/menu.lst), find any references to the old ramdisk image name, and replace them with the new name. 4. Reboot the system to verify the changes. 5. Install the Emulex lpfc Linux driver kit.
The Linux SCSI subsystem only sees 8 LUNs when more are present.	Some SCSI drivers will not scan past 8 LUNs when the target reports as a SCSI-2 device. Force SCSI bus scan with /usr/sbin/ lpfc/lun_scan. SuSE supplies /bin/rescan-scsi-bus.sh which can be changed to scan everything.
Cannot See Multiple Zones from the Management Server. Cannot see multiple zones on the same screen of my management server running the OneCommand Manager application.	Provide a physical FC connection into each of the zones. For each zone you want to see, connect an Emulex OneCommand Manager application-enabled port into that zone. Use Out-of-Band discovery, Ethernet, to connect to the undiscovered server.

Linux iSCSI

The following table provides Linux iSCSI troubleshooting information for the OneConnect UCNA.

Table 6: Linux iSCSI

Situation	Resolution
The operating system fails to install or you cannot successfully install the iSCSI or NIC drivers.	Verify that the operating system you are using is supported by OneConnect Server Software.
The OneConnect iSCSI BIOS banner is not displayed during system POST.	Go to your motherboard BIOS configuration and ensure Option ROM is enabled for the PCI-E slot into which the OneConnect UCNA is inserted.

Table 6: Linux iSCSI (Continued)

Situation	Resolution
Overall failure	Use the iSCSISelect utility to clear the Adapter Configuration. 1. From the Adapter menu, select **Clear Configuration**, then press <Enter>. 2. A message appears asking if you want to clear the current configuration. Press <Y>. 3. You are cautioned that the operation will remove any existing configuration permanently. Press <Y>. 4. After you clear the Adapter Configuration, reboot the system and then reconfigure the OneConnect UCNA.
The firmware fails to flash.	Use the CD-ROM ISO image located on CD2 to flash the firmware.
The iSCSI boot install fails.	• Verify the Boot target/LUN connectivity in iSCSISelect. • Check the system BIOS for boot device priority order.
The firmware becomes corrupted or non-responsive.	Update the firmware by using the Flash utility. To update the firmware, follow these steps: 1. Locate the ISO image file on CD2 and use it to create a bootable CD. 2. Boot to CD on a OneConnect UCNA-installed system. 3. Press <Y> when asked if you want to continue to update to the firmware version. The firmware automatically updates. 4. When complete, remove the flash CD from the CD drive, reboot, and verify that the BIOS banner shows the updated version.

Linux NIC

The following table provides Linux NIC troubleshooting information for the OneConnect UCNA.

Table 7: Linux NIC

Situation	Resolution
During boot, the system hangs after the OneConnect BIOS banner is displayed.	The firmware may be corrupted and may need to be reflashed. Update the firmware by using the Flash utility: 1. Locate the ISO image file on installation CD2 and create a bootable CD. 2. Boot to a CD on a OneConnect UCNA-installed system. 3. Press <Y> when asked if you want to continue to update to the firmware version. The firmware automatically updates. 4. When complete, remove the flash CD from the CD drive, reboot, and verify that the BIOS banner shows the updated version.
The driver works but the transmit and receive data rates are not near 10G/bs line rate.	There could be several reasons for poor performance. The driver logs a warning message if the card is found in a suboptimal slot. If you see this message, in /var/log/messages, move the card to the proper slot. For more information on getting the best performance from a OneConnect UCNA, see "Network Driver Performance Tuning" on page 25.

Linux Log Messages

Retrieving Linux NIC Warning Log Codes

Like all other driver and operating system messages, all be2net driver messages are logged in the file /var/log/messages. This log file is an ASCII text file and can be viewed and searched with a text editor such as vi. Large log files automatically rotate and rotated log files are named messages.x, where x is an integer.

To search the log file for error messages, at the command prompt, type:

```
# cd /var/log
# less messages
```

A message is displayed similar to the following:

```
Aug 15 09:57:48 S74 kernel: Invalid MTU requested. Must be between
256 and 8974 bytes
```

Linux NIC Warning Log Entries

The following is a list of Linux network driver warning log messages. When reporting a problem with the OneConnect UCNA to Emulex, check the kernel message log using the command dmesg or the file /var/log/messages and report any of these entries that may be present. All warning messages logged by OneConnect UCNA will start with “be2net <BID>”, where <BID> is the PCI bus identifier string. For example:

```
be2net 0000:0d:00.1: MTU must be between 256 and 9018 bytes.
```

Note: In the following table, <D>, <DD>, or <DDD> in the ‘Message Displayed’ column refers to decimal values that appear in the actual error messages.

Table 8: Linux NIC Warning Log Entries

Message Displayed	Description
`Could not get link status for eth<D>`	The firmware command to get the Link status returned an error.
`Could not set Rx buffer size to <DDD>. Using <DDD>`	The firmware command to change the RX buffer size failed. The driver still works with the default buffer size.
`MSIX Request IRQ failed - Errno <DDD>`	Request for MSIX interrupt registration failed. The driver still works with INTx interrupts.
`Unsupported receive buffer size (<DDD>) requested. Must be 2048 or 4096. Defaulting to 2048`	An unsupported receive buffer size was passed for module parameter rxbuf_size. The driver still works with a default RX buffer size of 2048.
`Failed to register char device`	Could not create the char device used for certain management functions. The driver must still work.
`Cannot support more than 2 BE Adapters`	The driver detected more than two OneConnect UCNAs in the system. The first two adapters are initialized and operate properly. The other devices will be ignored.

Table 8: Linux NIC Warning Log Entries (Continued)

Message Displayed	Description
`Unable to get BE Firmware Version`	The firmware command to get version number failed. The reason is most likely due to incompatible firmware.
`alloc_skb() failed`	Could not allocate an skb structure to pass to stack a frame received from the network. Transient failures can be ignored. Persistent messages point to a memory tight/leak problem.
`Invalid MTU requested. Must be between <DDD> and <DDD> bytes`	Invalid MTU size in MTU configuration ioctl. The MTU will not be changed.
`Unable to get pause frame settings`	The firmware command to get pause frame settings failed.
`Unable to set pause frame settings`	The firmware command to set pause frame settings failed.

Retrieving Linux iSCSI Warning Log Codes

For Linux systems, the OneConnect iSCSI (be2iscsi) driver generates warnings to the /var/log/messages log file. The log file is an ASCII text file and can be viewed and searched with your preferred text editor.

To search the log file for error messages, at the command prompt type:

```
# cd /var/log
# vim messages
```

For example, you may see the following message:

```
be2iscsi driver detected error 0x12790006
```

Linux iSCSI Error Log Code Entries

The following is a brief description of the error log codes generated by the OneConnect iSCSI Linux driver. It includes the error code, the message displayed, and the meaning of the message and the recommended resolution.

Table 9: Linux iSCSI Error Log Code Entries

Message ID	Message	Recommended Resolution
0x31880001	`The be2iscsi driver failed to load because initialization failed during a power management bootup.`	This failure may be due to the firmware not being present or running currently. This failure may also indicate a hardware problem.
0x3184000c	`The be2iscsi driver failed was unable to map one or more PCI Base Address Register and hence failed to load.`	This failure may indicate a low memory condition or a hardware error.

Table 9: Linux iSCSI Error Log Code Entries (Continued)

Message ID	Message	Recommended Resolution
0x3184000b	`The be2iscsi driver ignored a configuration entry since the entry was invalid.`	Check the registry configuration for any new entries added for Driver Parameters. The invalid entry needs to be removed or corrected. Refer to the driver readme file for the correct range of values.
0x31840006	`The be2iscsi driver failed to load due to memory allocation failure.`	This failure occurred due to a failed memory allocation in the driver. Check low memory conditions.
0x31840005	`The be2iscsi driver failed to load since it did not find the correct hardware IDs on the board.`	This failure indicates the OneConnect UCNA has an incorrect vendor ID, device ID, subsystem vendor ID, or subsystem device ID. Contact Emulex technical support.
0x31840001	`The be2iscsi driver failed to load because initialization failed during normal bootup.`	This failure may be due to the firmware not being present or running currently. This failure may also indicate a hardware problem.
0x31640004	`An internal API failed in be2iscsi driver during initialization.`	This failure may indicate a low memory condition.
0x14831000	`There was an Unrecoverable Error detected by the beiscsi driver. Following this error log entry, the next 3 entries will indicate the error codes.`	This may be due to hardware errors or due to unhandled exceptions in the hardware or firmware.
0x138e0103	`The be2iscsi driver failed an IOCTL request since the number of scatter gather elements required for the IOCTL buffer exceeded the BladeEngine’s firmware limit. Following this error log entry, the next entry will indicate the IOCTL opcode and the payload length requested.`	This error may indicate an incorrect configuration option for the OneConnect iSCSI driver. It may also indicate a low memory condition.

Table 9: Linux iSCSI Error Log Code Entries (Continued)

Message ID	Message	Recommended Resolution
0x138d0101	The be2iscsi driver detected an error during offloading the iSCSI connection. The operation will be retried again. Following this error log entry, the next entry will indicate the session handle and the BladeEngine firmware error code.	This may indicate a target is in error or may point to transient network connectivity issues. It may also indicate a OneConnect firmware error.
0x12990013	The be2iscsi driver did not receive an iSCSI command window update up to 25 seconds during I/O operations. Following this error log entry, the next entry will indicate the session handle where this error occurred. The beiscsi driver will trigger a session recovery on the session and continue.	Check for any errors reported at the target. The OneConnect iSCSI Initiator is only supported with certified Targets. Verify that the iSCSI Target is certified by Microsoft. Check for software updates at the target vendor’s Web site. Check for software updates at the Emulex Web site. If the above fails, contact Emulex technical support.
0x127b0012	The be2iscsi driver received an invalid iSCSI Command Sequence Number update from the target. Following this error log entry, the next three entries will indicate the session handle and the iSCSI parameters - MaxCmdSN and ExpCmdSN respectively.	Check for any errors reported at the target. The OneConnect iSCSI Initiator is only supported with certified Targets. Verify that the iSCSI Target is certified by Microsoft. Check for software updates at the target vendor’s website. Check for software updates at the Emulex Web site. If the above fails, contact Emulex technical support.
0x12790006	A connection to the target was lost for a period exceeding the Extended Timeout (ETO). The error log entry immediately following this entry will indicate the session ID of the target that lost the connection. There will be event log entries from the disk subsystem indicating that the drives were lost. If any I/Os were in progress, the system may see I/O errors or failures.	Check the connection to the target or the state of the target device. If the target is made available, any sessions that existed previously will be reestablished and the devices will be available for I/O.

Table 9: Linux iSCSI Error Log Code Entries (Continued)

Message ID	Message	Recommended Resolution
0x11990007	The be2iscsi driver received a Task Management Function that is not supported and rejected this request. The error log entry immediately following this entry will indicate the TMF function code that was rejected.	The operating system version is not supported.
0x11940008	The be2iscsi driver received a Task Management Function Abort request for an I/O request that is not present with the driver.	This may indicate a slow connection to the target. Check network connectivity to the target for any errors.
0x11840002	The be2iscsi driver encountered a mismatched version of the firmware running on the board. This error may be followed by more error codes 0x31840001 or 0x31880001 indicating that the be2iscsi driver failed to load.	This failure indicates that the driver version that is running on the system does not match the version of the firmware flashed on the board. Fix this by running the installer from the desired version.
0x11840001	The be2iscsi driver detected a failure in the hardware during initialization. This error may be followed by more error codes 0x31840001 or 0x31880001 indicating that the be2iscsi driver failed to load.	This failure indicates that the hardware has not been initialized or is malfunctioning. This may also indicate that the firmware is not running correctly.
0x11800005	Both Port 0 and Port 1 links were down for a period exceeding the Link Down Timeout (LDTO). If the initiator has connection to the target, there will be event log entries from the disk subsystem indicating that the drives were lost. If any I/Os were in progress, the system may see I/O errors or failures.	Check the links to the OneConnect UCNA. If the link is reestablished, any sessions that existed previously will be reestablished and the devices will be available for I/O.
0x11800003	Both Port 0 and Port 1 links are down.	Check the links to the OneConnect UCNA.

LPFC Log Messages

Introduction

This section lists the log messages for the FC and FCoE driver.

LPFC error log messages go to /var/log/messages.

Message Log Example

The following is an example of a LOG message:

```
Jul  2 04:23:34 daffy kernel: lpfc 0000:03:06.0: 0:1305 Link Down
Event x2f2 received Data: x2f2 x20 x110
```

In the above LOG message:

- lpfc 0000:03:06.0: identifies the identifies the pci location of the particular lpfc hw port.
- 0: identifies Emulex adapter0.
- 1305 identifies the LOG message number.

Note: If the word 'Data:' is present in a LOG message, any information to the right of 'Data:' is intended for Emulex technical support/engineering use only.

Log Messages

elx_mes0217: Block sgl registration required DMAsize <reqlen> great than a page

DESCRIPTION: The request to post SGL pages does not fit on a page.
DATA: None
SEVERITY: Warning
LOG: LOG_INIT
ACTION: None required.

elx_mes0221: FAN timeout

DESCRIPTION: A link up event was received without the login bit set, so the driver waits E_D_TOV for the Fabric to send a FAN. If no FAN if received, a FLOGI will be sent after the timeout.
DATA: None
SEVERITY: Warning
LOG: LOG_DISCOVERY verbose
ACTION: None required. The driver recovers from this condition by issuing a FLOGI to the fabric.

elx_mes0237: Pending Link Event during Discovery: State <hba_state>

DESCRIPTION: Received link event during discovery. Causes discovery restart.
DATA: None
SEVERITY: Warning
LOG: LOG_DISCOVERY verbose
ACTION: None required unless problem persists. If persistent check cabling.

elx_mes0274: lpfc_nlp_put: ndlp:x%pusgmap:x%x refcnt:%d, void *)ndlp, ndlp->nlp_usg_map, atomic_read(&ndlp->kref.refcount)

DESCRIPTION:
DATA: None
SEVERITY: Warning
LOG: LOG_NODE
ACTION: None required.

elx_mes0275: lpfc_nlp_put: ndlp:x%pusgmap:x%x refcnt:%d, void *)ndlp, ndlp->nlp_usg_map, atomic_read(&ndlp->kref.refcount)

DESCRIPTION: A kref_put was called again after the node was already inactive.
DATA: None
SEVERITY: Warning
LOG: LOG_NODE
ACTION: None required.

elx_mes0276: lpfc_nlp_get: ndlp:x%pusgmap:x%x refcnt:%d, void *)ndlp, ndlp->nlp_usg_map, atomic_read(&ndlp->kref.refcount)

DESCRIPTION: A kref_get was attempted on a node that was being released.
DATA: None
SEVERITY: Warning
LOG: LOG_NODE
ACTION: None required.

elx_mes0277: lpfc_enable_node: ndlp:x%pusgmap:x%x refcnt:%d, void *)ndlp, ndlp->nlp_usg_map, atomic_read(&ndlp->kref.refcount)

DESCRIPTION: Enable node was attempted on an inactive node.
DATA: None
SEVERITY: Warning
LOG: LOG_NODE
ACTION: None required.

elx_mes0278: lpfc_enable_node: ndlp:x%pusgmap:x%x refcnt:%d, void *)ndlp, ndlp->nlp_usg_map, atomic_read(&ndlp->kref.refcount)

DESCRIPTION: Enable node was attempted on an inactive node.
DATA: None
SEVERITY: Warning
LOG: LOG_NODE
ACTION: None required.

elx_mes0280: lpfc_cleanup_node: ndlp:x%pusgmap:x%x refcnt:%d, void *)ndlp, ndlp->nlp_usg_map, atomic_read(&ndlp->kref.refcount)

DESCRIPTION: Node clean-up was attempted on a node that has already been marked for memory free.
DATA: None
SEVERITY: Warning
LOG: LOG_NODE
ACTION: None required.

elx_mes0281: lpfc_cleanup_node: ndlp:x%pusgmap:x%x refcnt:%d, void *)ndlp, ndlp->nlp_usg_map, atomic_read(&ndlp->kref.refcount)

DESCRIPTION: Node clean-up was called to prepare the node for release.
DATA: None
SEVERITY: Warning
LOG: LOG_NODE
ACTION: None required.

elx_mes0301: READ_SPARAM: no buffers

DESCRIPTION: The driver attempted to issue a READ_SPARAM mailbox command to the adapter, but there were no buffers available.
DATA: None
SEVERITY: Warning
LOG: LOG_MBOX verbose
ACTION: This message indicates: (1) Kernel virtual memory is depleted. Check that the system meets minimum RAM requirements for the Emulex Fibre Channel adapter. Try closing other applications to free some memory. (2) A possible driver buffer management problem. If the problem persists, report the error to Technical Support.

elx_mes0302: REG_LOGIN: no buffers

DESCRIPTION: The driver attempted to issue a REG_LOGIN mailbox command to the adapter, but there were no buffers available.
DATA: (1) Did, (2) flag
SEVERITY: Warning
LOG: LOG_MBOX verbose
ACTION: This message indicates: (1) Kernel virtual memory is depleted. Check that the system meets minimum RAM requirements for the Emulex Fibre Channel adapter. Try closing other applications to free some memory. (2) A possible driver buffer management problem. If the problem persists, report the error to Technical Support.

elx_mes0313: Ring <ringno> handler: unexpected Rctl <Rctl> Type <Type> received

DESCRIPTION: The Rctl/Type of a received frame did not match any for the configured masks for the specified ring.
DATA: None
SEVERITY: Warning
LOG: LOG_SLI verbose
ACTION: This error could indicate a software driver, firmware or hardware problem. Report these errors to Technical Support.

elx_mes0322: Ring <ringno> handler: unexpected completion IoTag <IoTag>

DESCRIPTION: The driver could not find a matching command for the completion received on the specified ring.
DATA: (1) ulpStatus, (2) ulpWord[4], (3) ulpCommand, (4) ulpContext
SEVERITY: Warning
LOG: LOG_SLI verbose
ACTION: This error could indicate a software driver or firmware problem. If problems persist report these errors to Technical Support.

elx_mes0328: Rsp Ring <ring number> error: IOCB Data:

DESCRIPTION: The firmware has returned an error for this IOCB.
DATA: (1) <iocb word[0]…iocb word[7]>, (2) <rsp word[0]…rsp[word[7]>
SEVERITY: Warning
LOG: LOG_SLI
ACTION: None required.

elx_mes0336 - Rsp Ring <ringno> error: IOCB

DESCRIPTION: An IOCB error has occurred on the specified ring.
DATA: (1) ulpWord[0], (2) ulpWord[1], (3) ulpWord[2], (4) ulpWord[3], (5) ulpWord[4], (6) ulpWord[5], (7) irsp+6, (8) irsp+7
SEVERITY: Warning
LOG: LOG_SLI verbose
ACTION: If the problem persists, check the targets. If the targets are okay, report the error to Technical Support.

elx_mes0351: Config MSI mailbox command failed, mbxCmd <u.mb.mbxComm>, mbxStatus <u.mb.mbxStatus>

DESCRIPTION: The mailbox command sent to the firmware to configure the adapter to use MSI-X has failed.
DATA: None
SEVERITY: Warning
LOG: LOG_MBOX
ACTION: Ensure the hardware platform supports MSI-X.

elx_mes0357: MSI-X interrupt with no EQE

DESCRIPTION: SLI-4 adapter interrupt on the slow path but there is no associated EQE.
DATA: None
SEVERITY: Warning
LOG: LOG_SLI
ACTION: None required.

elx_mes0358: MSI-X interrupt with no EQE

DESCRIPTION: SLI-4 adapter interrupt on the fast path but there is no associated EQE.
DATA: None
SEVERITY: Warning
LOG: LOG_SLI
ACTION: None required.

elx_mes0373: FCP complete error: status=<status> hw_status=<hw status>, total_data_specified=<total data transferred>, parameter=<rsp word[4]>, word3=<wcqe word 3>

DESCRIPTION: Logs the FCP failure. Status and parameter are equivalent to ulpStatus and ulpWord[4].
DATA: None
SEVERITY: Warning
LOG: LOG_SLI
ACTION: None required.

elx_mes0374: FCP complete with no corresponding cmdiocb: iotag <iocb iotag>

DESCRIPTION: There was no IOCB on the in-progress list that matched this iotag.
DATA: None
SEVERITY: Warning
LOG: LOG_SLI
ACTION: None required.

elx_mes0375: FCP cmdiocb not callback function iotag: <iocb iotag>

DESCRIPTION: The IOCB found for this iotag does not have a completion handler set in it.
DATA: None
SEVERITY: Warning
LOG: LOG_SLI
ACTION: None required.

elx_mes0378: No support for fcpi mode.

DESCRIPTION: Could not configure the port to run in FCP initiator mode.
DATA: None
SEVERITY: Warning
LOG: LOG_MBOX, LOG_SLI
ACTION: None required.

elx_mes0379: Feature Mismatch Data: <req ftr word2 hex> <req_ftr word3 hex> <cfg_enable_npiv> <max vpi hex>

DESCRIPTION: The features passed in to the driver as module parameters do not match what the firmware can do. Setting to default values.
DATA: None
SEVERITY: Warning
LOG: LOG_MBOX, LOG_SLI
ACTION: None required.

elx_mes0383: Error <rc> during scsi sgl post operation

DESCRIPTION: The SGL entries could not be registered with the adapter.
DATA: None
SEVERITY: Warning
LOG: LOG_MBOX, LOG_SLI
ACTION: Reset the adapter using hbacmd.

elx_mes0386: ELS complete with no corresponding cmdiocb: iotag <iotag>

DESCRIPTION: The completion that the ISR is handling cannot find a tag associated with the IOTAG.
DATA: None
SEVERITY: Warning
LOG: LOG_SLI
ACTION: None required.

elx_mes0421: MSI-X slow-path request_irq failed <rc>

DESCRIPTION: The kernel API to request an IRQ has failed.
DATA: None
SEVERITY: Warning
LOG: LOG_INIT
ACTION: Use module parameter lpfc_use_msi=0 (IntX).

elx_mes0429: MSI-X fast-path request_irq failed (<rc>)

DESCRIPTION: The driver received an error for the request_irq_call.
DATA: None
SEVERITY: Warning
LOG: LOG_INIT
ACTION: Unload and reload the driver.

elx_mes0458: Bring adapter online

DESCRIPTION: The FC driver has received a request to bring the adapter online. This may occur when running lputil.
DATA: None
SEVERITY: Warning
LOG: LOG_INIT verbose
ACTION: None required.

elx_mes0460: Bring adapter offline

DESCRIPTION: The FC driver has received a request to bring the adapter offline. This may occur when running lputil.
DATA: None
SEVERITY: Warning
LOG: LOG_INIT verbose
ACTION: None required.

elx_mes0466: Outstanding IO when bringing Adapter offline

DESCRIPTION: IO is still pending while attempting to stop the driver.
DATA: None
SEVERITY: Warning
LOG: LOG_INIT
ACTION: None required.

elx_mes0478: MSI request_irq failed (<rc>).

DESCRIPTION: The request_irq kernel API has failed.
DATA: None.
SEVERITY: Warning
LOG: LOG_INIT
ACTION: Set lpfc_use_msi=0.

elx_mes0485: MSI-X slow-path request_irq failed (<rc>).

DESCRIPTION: The request_irq kernel API has failed.
DATA: None
SEVERITY: Warning
LOG: LOG_INIT
ACTION: Set module parameter lpfc_use_msi=0.

elx_mes0486: MSI-X fast-path (<index>) request_irq failed (<rc>).

DESCRIPTION: The request_irq kernel API has failed.
DATA: None
SEVERITY: Warning
LOG: LOG_INIT
ACTION: Set module parameter lpfc_use_msi=0.

elx_mes0490: MSI request_irq failed (<rc>).

DESCRIPTION: The request_irq kernel API has failed.
DATA: None
SEVERITY: Warning
LOG: LOG_INIT
ACTION: Set module parameter lpfc_use_msi=0.

elx_mes0544: lpfc_create_static_vport failed to issue dump mailbox command ret <rc> status <mbxStatus>

DESCRIPTION: Failed to issue a dump mailbox command for static VPort creation.
DATA: None
SEVERITY: Warning
LOG: LOG_INIT
ACTION: None required.

elx_mes0546: lpfc_create_static_vport failed to create vport

DESCRIPTION: Failed to create a VPort.
DATA: None
SEVERITY: Warning
LOG: LOG_INIT
ACTION: None required.

elx_mes0704: At limitation of <total> preallocated command buffers

DESCRIPTION: The maximum number of command buffers have already been allocated.
DATA: None
SEVERITY: Warning
LOG: LOG_FCP verbose
ACTION: None required.

elx_mes0705: Allocation request of <num> command buffers will exceed max of <hba_queue_depth>. Reducing allocation request to <size>

DESCRIPTION: The number of command buffers requested will exceed the maximum so a smaller quantity will be allocated.
DATA: None
SEVERITY: Warning
LOG: LOG_FCP verbose
ACTION: None required.

elx_mes0708: Allocation request of <num_to_alloc> command buffers did not succeed. Allocated <num_allocated> buffers.

DESCRIPTION: The allocation request for the specified command buffers did not succeed. However, the specified number of buffers has been allocated.
DATA: None
SEVERITY: Warning
LOG: LOG_FCP
ACTION: None required.

elx_mes0711: detected queue full - lun queue depth adjusted to%d

DESCRIPTION: The driver detected a queue full status on a scsi command response. New lun queue depth is reported
DATA: (1) New lun queue depth
SEVERITY: Warning
LOG: LOG_FCP verbose
ACTION: This may indicate an oversubscribed target array. Check your SAN configuration and IO workload.

elx_mes0749: SCSI layer issued abort device

DESCRIPTION: The SCSI layer aborted a device.
DATA: (1) ret, (2) id, (3) lun, (4) snum
SEVERITY: Warning
LOG: LOG_FCP verbose
ACTION: None required.

elx_mes1259: mbox: Issued mailbox cmd <u.mb.mbxCommand> while in stopped state.

DESCRIPTION: Only the dump mailbox command and reset adapter mailbox command are allowed when in the stopped state.
DATA: None
SEVERITY: Warning
LOG: LOG_MBOX
ACTION: None required.

elx_mes1268: Find ndlp returned NULL for oxid:x%x SID:x%x, oxid, sid.(int)off, rc.

DESCRIPTION: Could not find the node for this DID.
DATA: None
SEVERITY: Warning
LOG: LOG_ELS
ACTION: None required.

elx_mes1302: Invalid speed for this board: Reset link speed to auto: <cfg_link_speed>

DESCRIPTION: The driver is reinitializing the link speed to auto-detect.
DATA: None
SEVERITY: Warning
LOG: LOG_LINK_EVENT verbose
ACTION: None required.

elx_mes1304: Link Up Event ALPA map

DESCRIPTION: A link up event was received.
DATA: (1) wd1, (2) wd2, (3) wd3, (4) wd4
SEVERITY: Warning
LOG: LOG_LINK_EVENT verbose
ACTION: If numerous link events are occurring, check the physical connections to the Fibre Channel network.

elx_mes1801 Create vport work array FAILED: cannot do scsi_host_get

DESCRIPTION: The driver was unable to get a reference to a SCSI host.
DATA: None
SEVERITY: Warning
LOG: LOG_VPORT verbose
ACTION: Software driver warning. If this problem persists, report these errors to Technical Support.

elx_mes1816 FLOGI NPIV supported, response data <port>

DESCRIPTION: The fabric reports support for NPIV upon FLOGI.
DATA: (1) response_multiple_NPort
SEVERITY: Warning
LOG: LOG_VPORT verbose
ACTION: No action needed, informational.

elx_mes1817 Fabric does not support NPIV - configuring single port mode

DESCRIPTION: The fabric reports no support for NPIV upon FLOGI.
DATA: None
SEVERITY: Warning
LOG: LOG_VPORT verbose
ACTION: No action needed, informational.

elx_mes2004: Failed to allocate XRI.last XRITAG is <XRI> Max XRI is <MAX_XRI>, Used XRI is <USED_XRI>.

DESCRIPTION: All XRIs are in use.
DATA: None
SEVERITY: Warning
LOG: LOG_SLI
ACTION: None required.

elx_mes2518: Requested to send 0 NOP mailbox cmd

DESCRIPTION:
DATA: None
SEVERITY: Warning
LOG: LOG_INIT
ACTION: None required.

elx_mes2520: NOP mailbox command failed status x%x add_status x%x mbx status x%x, shdr_status, shdr_add_status, rc.

DESCRIPTION: The NOP mailbox command has failed.
DATA: None
SEVERITY: Warning
LOG: LOG_INIT
ACTION: None required.

elx_mes2539: Dropped frame rctl:<“RCTL TYPE”> type:<“FH TYPE”>

DESCRIPTION: The receive frame has an unsupported RCTL or FH_TYPE.
DATA: None
SEVERITY: Warning
LOG: LOG_ELS
ACTION: None required.

elx_mes2540: Ring <ring #> handler: unexpected Rctl <fh_rctl> Type <fh_type>

DESCRIPTION: The received frame has an unsupported RCTL or FH_TYPE.
DATA: None
SEVERITY: Warning
LOG: LOG_SLI
ACTION: None required.

elx_mes2542: Try to issue mailbox command <vpi> (<mbxCommand>) synchronously ahead of async mailbox command queue

DESCRIPTION: Attempting to send a synchronous mailbox command ahead of the asynchronous mailbox commands.
DATA: (1) sli4_mbx_opcode, (2) sli_flag, (3) flag
SEVERITY: Warning
LOG: LOG_MBOX, LOG_SLI
ACTION: None required.

elx_mes2559: Block sgl registration required DMA size <reqlen> great than a page.

DESCRIPTION: Attempting to register more SGEs with the firmware than can fit in a page.
DATA: None
SEVERITY: Warning
LOG: LOG_INIT
ACTION: None required.

elx_mes2569: lpfc_dump_fcoe_param: memory allocation failed

DESCRIPTION: Memory allocation has failed.
DATA: None
SEVERITY: Warning
LOG: LOG_MBOX
ACTION: None required.

elx_mes2579: Slow-path wqe consume event carries miss-matched qid: wcqe-qid=<wcqe_quid>, sp-qid=<sp_quid>

DESCRIPTION: The consumed entry does not have the slow path's queueID.
DATA: None
SEVERITY: Warning
LOG: LOG_SLI
ACTION: None required.

elx_mes2580: Fast-path wqe consume event carries miss-matched qid: wcqe-qid=<fcp_wqid>.

DESCRIPTION: The consumed entry does not have the fast path's queueID.
DATA: None
SEVERITY: Warning
LOG: LOG_SLI
ACTION: None required.

elx_mes2582: Not enough WQs (<max_wq>) from the pci function for supporting the requested FCP WQs (<cfg_wq_count>), the actual FCP WQs can be supported: <wq_count>

DESCRIPTION: The driver was not configured with enough fast-path work queues.
DATA: None
SEVERITY: Warning
LOG: LOG_INIT
ACTION: None required.

elx_mes2593: The FCP EQ count(<cfg_fcp_eq_count>) cannot be greater than the FCP WQ count(<cfg_fcp_wq_count>), limiting the FCP EQ count to <cfg_fcp_wq_count>

DESCRIPTION: The fast-path event queue cannot be greater than the fast-path work queue count.
DATA: None
SEVERITY: Warning
LOG: LOG_INIT
ACTION: None required.

elx_mes2717: CT context array entry [<index>] over-run: oxid:<fh_ox_id>, sid:<fh_SID>

DESCRIPTION: All of the array slots to hold buffers that are passed to the application are in use.
DATA: None
SEVERITY: Warning
LOG: LOG_ELS
ACTION: None required.

elx_mes2718: Clear Virtual Link Received for VPI <index> tag <event_tag>

DESCRIPTION: A Clear virtual link was received from the Fabric for this VPI.
DATA: None
SEVERITY: Error
LOG: LOG_DISCOVERY
ACTION: None required.

elx_mes2719: Invalid response length: tgt <TGT_ID> lun <LUN> cmnd <CMD> rsplen <RSPLEN>

DESCRIPTION: The response length for this FCP command is not supported.
DATA: None
SEVERITY: Error
LOG: LOG_FCP
ACTION: None required.

elx_mes2721: ndlp null for oxid %x SID %x\n, icmd->ulpContext, dfchba->ct_ctx[tag].SID);

DESCRIPTION: The Node value for this SID is not in the node list.
DATA: None
SEVERITY: Warning
LOG: LOG_ELS
ACTION: None required.