

Emulex: An Insider's Guide to the Linux 2.6 Kernel

AT A GLANCE

The Linux 2.6 kernel brings Fibre Channel users a number of significant enhancements, many contributed to the operating system by Emulex, the maintainer of FC transport for the Linux kernel community. The latest SLES 10 and RHEL5 distributions, and the associated Emulex driver, introduce additional important features.

PRODUCTS

Linux 8.0 and 8.1 drivers and related management products

APPLICATIONS

This paper will assist users migrating from 2.4 kernel distributions, such as RHEL3 or SLES 8, to 2.6 kernel-based products, such as RHEL4 or SLES 9, and on to SLES 10 and RHEL5.

The Linux 2.6 kernel introduced a number of changes, including small computer system interface (SCSI) command scheduling and queuing, and automatic logical unit number (LUN) scanning. These changes, and a number of other 2.6 kernel innovations available in distributions such as Red Hat Enterprise Linux (RHEL) 4 and SUSE Linux Enterprise Server (SLES) 9, provide significant improvements over the 2.4 kernel and allow the Emulex 8.x driver implementation to evolve into a more robust driver focused on providing a solid Fibre Channel (FC) transport underneath the 2.6 kernel's enhanced SCSI subsystem.

The newest Linux distributions such as SLES 10 and RHEL5, based on the latest of the 2.6 kernels, also bring important manageability enhancements under the "sysfs" umbrella. Appendix 3 of this document illustrates the wealth of information available using sysfs. These enhancements have eliminated the need for an IOCTL add-on module with RHEL 5 and SLES 10 SP1, making user installation and deployment easier.

Operating systems are becoming more storage-competent

Most operating systems, including Linux, were initially designed with direct-attached SCSI or IDE/ATA storage in mind. The role of a Fibre Channel driver was to manage all the complexities brought by FC-based storage networking and sophisticated storage arrays, and present the operating system with a simplified view of a number of SCSI strings. The driver had to provide five main functions:

- ▶ Interface with the physical adapter (which manages the physical and lower FC protocol layers).
- ▶ Timing and flow control management; very different for networked FC arrays from local SCSI drives. This includes queue management and timers.
- ▶ Hiding FC discovery and temporary device loss. The driver must make the FC fabric appear as stable and static as the older parallel SCSI bus.

- ▶ Management of a number of error cases.
- ▶ Reporting of a number of administrative parameters (configuration, runtime events).

The remarkable growth in FC adoption has resulted in development of more sophisticated storage management by most operating systems, including Linux with the 2.6 kernel and the addition of the Fibre Channel transport in the SCSI subsystem. This is leading to more functions being handled by the SCSI mid-layer and thinner, more compact FC drivers.

Linux 2.6 kernel changes

The Linux 2.6 kernel was released by the Linux community via the kernel.org website in late 2003, and is now part of a growing number of distributions, including SUSE/Novell (SLES 9), Red Hat (RHEL 4), Oracle Enterprise Linux and Asianux. This kernel introduced significant changes that affected all I/O drivers that include Fibre Channel transports. Of particular note is a reworked SCSI layer, enhanced I/O scheduler and queue management, new sysfs interface for status and event collection, and driver-agnostic persistent naming modules. Four of the driver's five functions identified in the previous paragraph are affected:

- ▶ **Physical adapter interface:** this is the least affected, except by PHP (the PCI Hot Plug capability, which Emulex supports on all current adapters).
- ▶ **Timing and flow control management:** this driver function is drastically scaled back as these functions are properly handled in the SCSI subsystem.
- ▶ **Error management:** all Fibre Channel transport error cases continue to be handled by the driver, but SCSI error cases (discussed later in this document) are now handled by the SCSI subsystem.
- ▶ **Administrative reporting/management interface:** the events and parameters reported are largely unchanged, but they are now exposed through a new Linux device driver interface, 'sysfs', making them more accessible to third party applications as well as user-developed scripts and tools. The Emulex implementation of sysfs enables dynamic logging without reboot or driver reload, which is of critical value to enable storage and fabric troubleshooting.

The positive aspects of this change are very visible:

- ▶ Unified architecture spanning distributions and specific drivers.
- ▶ Convergence of vendor-qualified enterprise drivers and kernel.org and distribution drivers (subject to additional enterprise kitting for utilities and libraries).
- ▶ Consistent interface for third party applications.
- ▶ Reduced testing and qualification complexity for each individual driver or device.

Emulex has taken a leadership role in Linux FC implementation:

- ▶ Emulex sponsored the Fibre Channel transport patch that provided target block/unblock, fc_host, fc_rport (remote port), and target rescan capabilities for lower layer device drivers (LLDDs) that implement a Fibre Channel transport; the Emulex sponsored transport is vendor-agnostic and is currently implemented in the QLogic and IBM kernel drivers. The Fibre Channel transport is part of kernels 2.6.11 and higher. Emulex is implementing the Common HBA API (SNIA API) as well as its own command line and graphical management tools on top of the Linux sysfs system, the community-recommended method for managing devices in the 2.6 environment. The sysfs system eliminates the need for proprietary IOCTL's, makes the driver easier to review and maintain, and allows software vendors and end users to easily access Fibre Channel configuration and status information with the tools of their choice.
- ▶ Emulex is actively testing and providing production-enhancing requests to the developers of Device Mapper for its multipath I/O capabilities (dm_multipath)

Emulex 2.6 kernel driver strategy

Emulex has worked as a leading member of the Linux/ SCSI community to deliver important features that are now part of the 2.6 kernel:

- ▶ SCSI Midlayer Extensions (SCSI device block/unblock to cover cable pulls, the addition of transport objects, SCSI netlink event interface, miscellaneous utility functions)
- ▶ Fibre Channel Transport (first major transport implementation used as template for other transports,

FC Remote Port topology display in sysfs without custom tools, vendor-agnostic management interface including Host, Rport, and Target class objects with attributes and statistics based on HBA API, transport function library for cable pulls, consistent target ID bindings and event posting).

- ▶ The same Linux 8.x HBA driver distributed through Emulex and OEMs for production, also part of Linux kernels starting from 2.6.12rc3 .
- ▶ The Linux kernel maintainer for Fibre Channel transport is an Emulex engineer and architect whose time is donated to the community to that effect.

The latest driver updates that are part of the Linux kernel are then brought into a variety of Linux distributions. Emulex further collaborates with Red Hat, Novell and Asianux to provide enterprise-class qualification and support to RHEL, SLES, Red Flag, Haansoft and Miracle Linux.

Emulex will continue its policy of providing close direct support to its tier-1 OEMs, and of providing kits extending and completing our product beyond the confines of standard kernel components: management libraries, adapter diagnostic utilities, graphical management applications, etc.

Emulex, in working with maintainers of the Linux operating system and SCSI midlayer, has made modifications designed to create consistent, vendor-agnostic behavior in many areas, leverage the advances within the 2.6 kernel and, in the end, to create a more robust Fibre Channel Protocol (FCP) driver that takes full advantage of 2.6 kernel capabilities. As a result of these changes, much functionality has been transferred out of the device driver, deferring instead to kernel facilities that provide similar or identical functions. These differences are described below. Note that the Emulex 8.x series driver continues to provide:

- ▶ SCSI/FCP Support.
- ▶ Boot support on all supported architectures (x86, x64, ia64 and ppc64).
- ▶ Driver support for the Emulex HBA API (v1) library.
- ▶ Driver support for Emulex utilities including lputil, HBAnyware and AutoPilot Manager.

Another important feature is the ability provided by the Emulex driver to add new LUNs and targets without a reboot. When the devices are connected, configured on the fabric and ready to be inserted, a simple command will add them to the server configuration.

The Emulex Linux driver and host bus adapters support the enhanced 2.6 kernel PHP capability. In addition, the 8.1 driver offers functional extensions including support for Message-Signaled Interrupt (MSI*) in addition to polled mode and soft World Wide Port Name/World Wide Node Name (WWPN/WWNN) setting.

Detailed review of changes to the 2.6 interface and Emulex 8.x driver:

1 The driver defers all LUN handling to the SCSI midlayer.

In the 2.6 kernel, the SCSI midlayer handles management of all storage devices, including SCSI LUNs. The philosophy of LUN management is different, with user-controlled driver parameters now replaced by system-controlled dynamic variables:

- ◇ **lpfc_lun_skip:** Retired from the driver implementation. The driver now defers to the midlayer's whitelist/blacklist for devices and the BLIST_SPARSELUN flag. Target devices that do not complete SCSI device discovery properly because of compliance issues with SCSI-2 or SCSI-3 are now required to use the BLIST database to correct SCSI device discovery.
- ◇ **lpfc_max_luns:** Specifies the maximum number of LUNs per target supported by the driver. This parameter no longer sizes tables internal to the driver. The value is provided to the midlayer, where it is used as a maximum value for probing and adding LUNs during report LUNs and sequential scans. The default value is 256. This parameter is global, thus affecting all ports, and is not writable after driver load.
- ◇ **lpfc_dqfull_throttle_up_inc, lpfc_dqfull_throttle_up_time:** These little-used variables have been retired at the SCSI maintainer request. The kernel does not export a routine to alter the lun_queue_depth based on queue full return values from the target, but instead defers to the SCSI subsystem to manage the queue full responses.

* For more information on MSI or Message Signaled Interrupt please refer to a paper located on the web at <http://lwn.net/Articles/44139/>.

- ◇ **lpfc_lun_queue_depth:** Specifies the maximum number of outstanding SCSI commands per LUN. This number defaults to 30 in the driver and is communicated to the midlayer for its LUN queue management. This parameter is global, thus affecting all ports, and must be set before driver load. In addition, the Emulex 2.6 kernel driver now takes advantage of all the midlayer queue depth handling and queue full handling policy. This provides more consistent behavior among SCSI products.

2 The driver no longer sets per-target management parameters:

- ◇ **lpfc_max_target:** Retired from the driver implementation. This parameter is used to specify the maximum number of targets supported per HBA. This number has become a fixed value of 256 in the driver; the choice of 256 is to accommodate JBODs rather than target arrays, since production environments where 10 or more target arrays attached to the same server initiator, are uncommon.
- ◇ **lpfc_tgt_queue_depth:** Retired from the driver implementation. The 2.6 midlayer's LUN queue depth handling and queue full handling made this parameter redundant.

3 Per-adapter settings are not supported via modprobe.conf.

Modprobe.conf is an operating system–provided file, as opposed to an Emulex driver file. Driver configuration at boot time, when defined by modprobe.conf, is global for all Emulex HBAs. You retain the option of changing per-port settings for dynamic parameters after initial boot.

The Emulex default global configuration for boot purposes continues to support auto-topology and auto-speed configuration as a default, thus accommodating the most common per-adapter differences automatically.

4 The SCSI midlayer performs all SCSI error recovery.

This provides consistent error handling across all transports binding to the SCSI layer, and consistent error messages provided to relevant applications. As a result, the following configuration variables are no longer supported in the 8.x driver:

- ◇ **lpfc_check_cond_err:** Specifies the treatment of certain check conditions as SCSI–FCP response (RSP) errors; Emulex does not believe this parameter is currently in use.
- ◇ **lpfc_delay_rsp_err:** Specifies whether the driver delays SCSI–FCP response errors and certain check conditions before returning them to the SCSI layer; Emulex is not aware of any vendor currently making use of this parameter.
- ◇ **lpfc_extra_io_tmo:** Specifies an extra timeout value added to the individual SCSI–FCP command's operating system assigned timeout; retry logic and Fibre Channel block semantics in the SCSI midlayer make this parameter obsolete.
- ◇ **lpfc_no_device_delay:** Specifies the time interval, in seconds, between detecting a failed I/O and actually failing the I/O resulting from a Fibre Channel target device loss event; retry logic and Fibre Channel block semantics in the SCSI midlayer make this parameter obsolete.
- ◇ **lpfc_nodev_holdio:** Specifies whether the driver holds I/O errors if a device disappears. The hold is cancelled when the device returns; retries and the Fibre Channel block semantics in the SCSI midlayer make this parameter obsolete.

With the removal of these parameters, and the enhancements in the midlayer's error handling capabilities, comes some new behavior in the 2.6 Linux kernel with SCSI device (LUN) state. Distributions based on the 2.6.9 kernel introduced a SCSI device attribute called 'state' that represents that device's current operational state. For example, a device could be running, blocked or offline. These states are new to 2.6.9+ in particular and represent enhanced error handling by the SCSI midlayer.

Of all the new states, the offline state has caused the most confusion and concern for internal qualification efforts and for external customers. A SCSI device (LUN) transitions to the offline state whenever an error has occurred on a LUN from the midlayer's point of view, but only after various retry attempts have been made

(including basic abort/retry, LUN resets, and bus resets). When the decision to move to the offline state is made, the midlayer's error handling takes the device offline, forcing a system administrator to get involved. Once the problem is cleared, the system administrator can put that device back online either manually or by using an Emulex script called `lun_change_state.sh`.

Distributions using kernels preceding 2.6.9 have very similar behavior, but they don't have this state attribute. These distributions are also correctable via the `lun_change_state.sh` script.

5 New parameters in all 2.6 kernel based drivers including RHEL4, SLES 9, SLES 10, RHEL5 (did not exist in 2.4 kernel based drivers):

- ◇ **lpfc_hba_queue_depth**: Specifies the limit for the number of outstanding commands that may be issued to the adapter, and is used to manage adapter I/O queue depth. It is initially assigned to the adapter maximum. The default value is 8192. In cases where it must be tuned, you can set it to a lower value. For example, to avoid overloading specific target device I/O limits, you can set the value to 1024. Note that this limit will apply across all targets and all LUNs seen by the adapter.
- ◇ **issue_reset**: This write-only parameter is used for atomically issuing HBA resets.
- ◇ **board_mode**: This parameter specifies the state of the HBA (online, offline, warm, error).

6 Parameters that exist on RHEL4/SLES 9 Emulex drivers but have been retired from the SLES 10/RHEL5 drivers:

- ◇ **lpfc_linkdown_tmo**
- ◇ **lpfc_fcp_bind_method**
- ◇ **lpfc_discovery_min_wait**
- ◇ **lpfc_discovery_wait_limit**
- ◇ **disc_npr**
- ◇ **disc_map**
- ◇ **disc_unmap**

- ◇ **disc_pri**
- ◇ **disc_regl**
- ◇ **disc_adisc**
- ◇ **disc_plogi**

7 Persistent Naming—formerly called Persistent Binding—is managed by the 2.6 Linux hotplug subsystem. The 2.6 kernel includes two primary mechanisms for persistent naming:

- ◇ **udev** (described at http://www.kroah.com/linux/talks/ols_2003_udev_paper/Reprint-Kroah-Hartman-OLS2003.pdf).
- ◇ **devlabel** (described at <http://linux.dell.com/devlabel/devlabel.html>).

Emulex recommends udev for operation with the 2.6 kernel because it is the mainstream choice, although the Emulex driver operates with both udev and devlabel.

The introduction of udev into the 2.6 kernel stream represents a major function change to the Emulex 8.x driver. The driver no longer provides persistent bindings as it did in 7.x because it defers to the distribution's udev implementation. Udev can run at boot time for boot-from-SAN configurations or during OS load time. Either configuration provides persistent naming for SCSI devices in the SAN. Given a SCSI device with a unique identifier, udev provides system administrators with the ability to associate a name with that SCSI device and have it persist across reboots.

The udev mechanism provides binding at the device/LUN level. This resolves the issue that LUN-reordering posed to previous target ID binding methods.

8 The 8.x driver no longer handles Peripheral Set Addressing (PSA) or Volume Set Addressing (VSA) for LUN addresses. The SCSI midlayer now handles these addressing modes during the post report LUNs processing. This provides a common, vendor-agnostic mechanism for management of this function. Reference for VSA addressing: SCSI Architecture Model -2 section 4.9.3—Single Level logical Unit Numbers (address methods 00b and 01b).

9 Link Event Management. In the RHEL5 and SLES 10 kernels, and in the corresponding Emulex drivers, while the driver still responds to and provides messaging for link events, the driver no longer manages a link-down timeout value. Therefore, the driver has the following changes:

- ◇ **lpfc_linkdown_tmo:** Retired from the driver implementation. The link-down functionality has been moved into the per-target timeout value. The new parameter is called `no_dev_tmo` and is manageable at driver load time and during runtime.

10 Discovery Changes. With the retiring of the `lpfc_linkdown_tmo` parameter mentioned above, the driver focused its event processing around the disappearance and reappearance of targets. The driver now provides a timeout value for targets to disappear and recover before returning connect errors to the SCSI midlayer:

- ◇ **lpfc_nodev_tmo:** Specifies the amount of time, in seconds, a discovered target can disappear before the driver treats it as no connection. The current default is 30 seconds. The timeout value is global to all targets managed on a particular host instance, but is not global to all ports in a system.

To aid in faster initial discovery times and larger SAN events involving multiple targets, the driver modified the discovery threads parameter:

- ◇ **lpfc_discovery_threads:** Specifies the number of outstanding ELS events the driver allows at any time. The new default is 32. Earlier programs, such as 7.x, defaulted this value to 1. With the default setting of 32, it is likely that targets will be discovered in a different order from boot to boot. As a result, udev is highly encouraged to maintain target/LUN persistent naming. You can divert to the former, slower, discovery by setting this variable down to 1.

To simplify the type of binding to nodes, the driver condensed the many bind methods into a single parameter:

- ◇ **lpfc_fcp_bind_method:** Specifies WWNN, WWPN, DID or ALPA method of binding for each `N_Port`; consistent bindings and mapped bindings use this method.

11 Boot from SAN. Much of the work required to build a boot-from-SAN configuration remains the same as in previous driver programs. What is different is the use of udev in the initial random-access memory (RAM) disk to provide the persistent name for mapping the unique LUN identifier on a storage device to a name that persists across reboots in the initiator.

The use of udev in the initial RAM disk caused problems for Linux users on some Linux distributions. The problem was tracked down to an incomplete file set used to compile the initial RAM disk—specifically, some of the more important udev files, such as some rules files and the applications necessary to execute the rules, were missing. Also, particular to Fibre Channel, the new Fibre Channel transport module was missing, causing driver load errors in boot-from-SAN configurations.

The distributions that had this problem have been repaired with later updates.

About the Emulex 8.0 and 8.1 Drivers

The Emulex 8.0 and 8.1 drivers are Open Source drivers published under the GNU General Public License (“GNU GPL” or “GPL”) under the same terms as other driver versions. Applications and libraries are published under the Emulex license. Driver source code is published and available for inspection on the sourceforge web site (<http://sourceforge.net/projects/lpfcxxx>). Versions of the 8.0/8.1 drivers are also included in the Linux kernel (starting with 2.6.12 RC3) and distributed as part of Red Hat Enterprise Linux 4 and 5, as well as a number of other 2.6 based distributions not qualified by Emulex.

Emulex also provides on its web site the HBAnyware management application, which allows users to manage HBAs and connectivity either locally on a server or remotely. Capabilities range from power-on self-test (POST) and other HBA diagnostics, to parameter settings (either for individual HBAs or globally for a criteria-based population of defined set of HBAs) and finely tunable Fibre Channel event logging capabilities capturing host, fabric and array-generated events and incidents for fast and non-disruptive troubleshooting.

Appendix 1—8.x Driver Parameter List

1. Common module parameters for LPFC 8.0 (RHEL4, SLES 9) and 8.1 (RHEL5, SLES 10) drivers

Parameter	Access Rights	Avail. via sysfs	Run-time/Load-time Configurable
lpfc_fdmi_on	RW	Yes	Run-time
lpfc_log_verbose	RW	Yes	Run-time
lpfc_nodev_tmo	RW	Yes	Run-time
lpfc_use_adisc	RW	Yes	Run-time
lpfc_scan_down	RW	Yes	Run-time
serialnum	R	Yes	Run-time/Status
fwrev	R	Yes	Run-time/Status
hdw	R	Yes	Run-time/Status
lpfc_drvr_version	R	Yes	Run-time/Status
info	R	Yes	Run-time/Status
lpfc_scsi_req_tmo	RW	No	Load-time
lpfc_cr_count	RW	No (8.0)	Load-time
		Yes (8.1)	
lpfc_cr_delay	RW	No (8.0)	Load-time
		Yes (8.1)	
lpfc_ack0	R	Yes	Load-time
lpfc_fcp_class	R	Yes	Load-time
lpfc_link_speed	R	Yes	Load-time
lpfc_lun_queue_depth	R	Yes	Load-time
lpfc_max_luns	R	Yes	Load-time
lpfc_scan_down	R	Yes	Load-time
lpfc_topology	R	Yes	Load-time
lpfc_discovery_threads	R	No	Load-time

Accessible via sysfs;
run-time (dynamic)
configurable

Accessible via sysfs;
load-time (static)
configurable only

Not accessible via sysfs;
load-time (static)
configurable only

2. LPFC 8.0 (RHEL4, SLES 9) ONLY driver module parameters

Parameter	Access Rights	Avail. via sysfs	Run-time/Load-time Configurable
board_online	RW	Yes	Run-time
issue_lip	WO	Yes	Run-time
node_name	R	Yes	Run-time
port_name	R	Yes	Run-time
port_type	R	Yes	Run-time
portfcid	R	Yes	Run-time
disc_npr	R	Yes	Run-time
disc_map	R	Yes	Run-time
disc_unmap	R	Yes	Run-time
disc_prli	R	Yes	Run-time
disc_regln	R	Yes	Run-time
disc_adisc	R	Yes	Run-time
disc_plogi	R	Yes	Run-time
Ffabric_name	R	Yes	Run-time
lpfc_fcp_bind_method	R	Yes	Load-time
lpfc_linkdown_tmo	RW	Yes	Load-time
lpfc_discovery_min_wait	RW	Yes	Load-time
lpfc_discovery_wait_limit	RW	Yes	Load-time
lpfc_pci_max_read	RW	Yes	Load-time
lpfc_soft_wwpn_enable	WO	Yes	Load-time

Accessible via sysfs;
run-time (dynamic)
configurable

Accessible via sysfs;
load-time (static)
configurable only

3. LPFC 8.1 (RHEL5, SLES 10) ONLY driver module parameters

Parameter	Access Rights	Avail. via sysfs	Run-time/Load-time Configurable
board_mode	RW	Yes	Run-time
lpfc_soft_wwn_enable	WO	Yes	Run-time
issue_reset	WO	Yes	Run-time
lpfc_soft_wwn	RW	Yes	Run-time
lpfc_devloss_tmo	RW	Yes	Load-time
lpfc_poll	RW	Yes	Load-time
lpfc_poll_tmo	RW	Yes	Load-time
lpfc_multi_ring_rctl	R	Yes	Load-time
lpfc_multi_ring_support	R	Yes	Load-time
lpfc_multi_ring_type	R	Yes	Load-time

Accessible via sysfs;
run-time (dynamic)
configurable

Accessible via sysfs;
load-time (static)
configurable only

Appendix 2—Driver Attributes Removed from 8.x Driver

1. The following parameters, available with current 7.x drivers for the 2.4 kernel, have been removed from 8.x drivers for the 2.6 kernel

7.x Parameter	2.6 Kernel Handling	7.x Parameter	2.6 Kernel Handling
lpfc_check_cond_err	Managed by midlayer	lpfc_linkdown_tmo	[Driver] lpfc_nodev_tmo
lpfc_delay_rsp_err		lpfc_automap	Automap always
lpfc_dqfull_throttle_up_time		lpfc_inq_pqb_filter	Prior distribution workaround
lpfc_dqfull_throttle_up_inc		lpfc_network_on	IP/FC removed
lpfc_extra_io_tmo		lpfc_ip_class	
lpfc_lun_skip		lpfc_post_ip_buf	
lpfc_no_device_delay		lpfc_xmt_queue_size	
lpfc_nodev_holdio		lpfc_max_target	N/A
lpfc_fcp_bind_DID	Replaced by one parameter: lpfc_fcp_bind_method	lpfc_tgt_queue_depth	
lpfc_fcp_bind_WWN			
lpfc_fcp_bind_WWPN			

Appendix 3—HBA Management information available through sysfs

The Emulex Linux lpfc driver provides a rich set of information about the SCSI and Fibre Channel environment, including information about SCSI hosts, SCSI devices, Fibre Channel N_Ports and the Fibre Channel transport. This is all accomplished using the Linux sysfs interface, which enables logging of FC events (in the Emulex implementation, sysfs logging can be turned on or modified at run time without reboot or driver reload). The interface is scriptable like the /proc/scsi interface, but contains much more information as compared to /proc/scsi interface. The following are examples of the information presented using /sysfs/.

SCSI host information

```

/sys/class/scsi_host/host3/board_mode ==> online
/sys/class/scsi_host/host3/board_online ==> 1
/sys/class/scsi_host/host3/cmd_per_lun ==> 3
/sys/class/scsi_host/host3/ctlreg ==> (Sysfs file used for reading HBA Config registers.)
/sys/class/scsi_host/host3/fwrev ==> 1.91A5 (H2F1.91A5)
/sys/class/scsi_host/host3/hdw ==> 2003806d
/sys/class/scsi_host/host3/host_busy ==> 0
/sys/class/scsi_host/host3/info ==> Emulex LP9802 2Gb PCI-X Fibre Channel Adapter on PCI bus 02 device 28 irq 201
/sys/class/scsi_host/host3/issue_reset ==> (File used for resetting HBA)
/sys/class/scsi_host/host3/lpfc_ack0 ==> 0
/sys/class/scsi_host/host3/lpfc_cr_count ==> 1
/sys/class/scsi_host/host3/lpfc_cr_delay ==> 0
/sys/class/scsi_host/host3/lpfc_drvr_version ==> Emulex LightPulse Fibre Channel SCSI driver 8.1.6.5
/sys/class/scsi_host/host3/lpfc_fcp_class ==> 3
/sys/class/scsi_host/host3/lpfc_fdm_i_on ==> 0
/sys/class/scsi_host/host3/lpfc_hba_queue_depth ==> 1024
/sys/class/scsi_host/host3/lpfc_link_speed ==> 0
/sys/class/scsi_host/host3/lpfc_log_verbose ==> 0x0
/sys/class/scsi_host/host3/lpfc_lun_queue_depth ==> 30
/sys/class/scsi_host/host3/lpfc_max_luns ==> 255
/sys/class/scsi_host/host3/lpfc_multi_ring_support ==> 1
/sys/class/scsi_host/host3/lpfc_nodev_tmo ==> 30
/sys/class/scsi_host/host3/lpfc_poll ==> 0x0
/sys/class/scsi_host/host3/lpfc_poll_tmo ==> 10
/sys/class/scsi_host/host3/lpfc_scan_down ==> 1
/sys/class/scsi_host/host3/lpfc_topology ==> 0
/sys/class/scsi_host/host3/lpfc_use_adisc ==> 0
/sys/class/scsi_host/host3/management_version ==> 0.0.0
/sys/class/scsi_host/host3/mbx ==> (File used for sending HBA management mailbox commands)
/sys/class/scsi_host/host3/modeldesc ==> Emulex LP9802 2Gb PCI-X Fibre Channel Adapter
/sys/class/scsi_host/host3/modelname ==> LP9802
/sys/class/scsi_host/host3/nport_evt_cnt ==> 5
/sys/class/scsi_host/host3/num_discovered_ports ==> 3
/sys/class/scsi_host/host3/scan ==> (File used for scanning new storage)
/sys/class/scsi_host/host3/serialnum ==> MM41937498
/sys/class/scsi_host/host3/sg_tablesize ==> 64
/sys/class/scsi_host/host3/state ==> Link Up - Ready: Fabric
/sys/class/scsi_host/host3/unchecked_isa_dma ==> 0
/sys/class/scsi_host/host3/unique_id ==> 1
/sys/class/fc_host/host3/active_fc4s ==> 0x00 0x00 0x01 0x00 0x00 0x00 0x00 0x00 0x01 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
/sys/class/fc_host/host3/fabric_name ==> 0x1000006069519369
/sys/class/fc_host/host3/issue_lip ==> (File used for resetting link)
/sys/class/fc_host/host3/maxframe_size ==> 2048 bytes
/sys/class/fc_host/host3/node_name ==> 0x20000000c93dc171
/sys/class/fc_host/host3/port_id ==> 0x010400

```

```

/sys/class/fc_host/host3/port_name ==> 0x1000000c93dc171
/sys/class/fc_host/host3/port_state ==> Online
/sys/class/fc_host/host3/port_type ==> NPort (fabric via point-to-point)
/sys/class/fc_host/host3/speed ==> 2 Gbit
/sys/class/fc_host/host3/supported_classes ==> Class 3
/sys/class/fc_host/host3/supported_fc4s ==> 0x00 0x00 0x01 0x00 0x00 0x00 0x00 0x00 0x01 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
/sys/class/fc_host/host3/supported_speeds ==> 1 Gbit, 2 Gbit
/sys/class/fc_host/host3/symbolic_name ==> Emulex LP9802 FV1.91A5 DV8.1.6.5
/sys/class/fc_host/host3/tgtid_bind_type ==> wwwpn (World Wide Port Name)

```

Fibre channel statistics

```

/sys/class/fc_host/host3/statistics/dumped_frames ==> 0xffffffffffff
/sys/class/fc_host/host3/statistics/error_frames ==> 0x0
/sys/class/fc_host/host3/statistics/fcp_control_requests ==> 0x0
/sys/class/fc_host/host3/statistics/fcp_input_megabytes ==> 0x0
/sys/class/fc_host/host3/statistics/fcp_input_requests ==> 0x0
/sys/class/fc_host/host3/statistics/fcp_output_megabytes ==> 0x0
/sys/class/fc_host/host3/statistics/fcp_output_requests ==> 0x0
/sys/class/fc_host/host3/statistics/invalid_crc_count ==> 0x0
/sys/class/fc_host/host3/statistics/invalid_tx_word_count ==> 0xc
/sys/class/fc_host/host3/statistics/link_failure_count ==> 0x0
/sys/class/fc_host/host3/statistics/lip_count ==> 0x0
/sys/class/fc_host/host3/statistics/loss_of_signal_count ==> 0x0
/sys/class/fc_host/host3/statistics/loss_of_sync_count ==> 0x7
/sys/class/fc_host/host3/statistics/nos_count ==> 0x0
/sys/class/fc_host/host3/statistics/prim_seq_protocol_err_count ==> 0x0
/sys/class/fc_host/host3/statistics/reset_statistics ==>
/sys/class/fc_host/host3/statistics/rx_frames ==> 0x191
/sys/class/fc_host/host3/statistics/rx_words ==> 0x22900
/sys/class/fc_host/host3/statistics/seconds_since_last_reset ==> 0x73d
/sys/class/fc_host/host3/statistics/tx_frames ==> 0x4bf
/sys/class/fc_host/host3/statistics/tx_words ==> 0x4f00

```

Information about all Fibre Channel NPorts (initiator + target) visible to the HBA

```

/sys/class/fc_remote_ports/rport-3:0-0/dev_loss_tmo ==> 35
/sys/class/fc_remote_ports/rport-3:0-0/maxframe_size ==> 2112 bytes
/sys/class/fc_remote_ports/rport-3:0-0/node_name ==> 0x1000006069519369
/sys/class/fc_remote_ports/rport-3:0-0/port_id ==> 0xfffffe
/sys/class/fc_remote_ports/rport-3:0-0/port_name ==> 0x2004006069519369
/sys/class/fc_remote_ports/rport-3:0-0/port_state ==> Online
/sys/class/fc_remote_ports/rport-3:0-0/roles ==> Fabric Port
/sys/class/fc_remote_ports/rport-3:0-0/scsi_target_id ==> -1
/sys/class/fc_remote_ports/rport-3:0-0/supported_classes ==> Class 2, Class 3
/sys/class/fc_remote_ports/rport-3:0-1/dev_loss_tmo ==> 35
/sys/class/fc_remote_ports/rport-3:0-1/maxframe_size ==> 2048 bytes
/sys/class/fc_remote_ports/rport-3:0-1/node_name ==> 0x1000006069519369
/sys/class/fc_remote_ports/rport-3:0-1/port_id ==> 0xfffffc
/sys/class/fc_remote_ports/rport-3:0-1/port_name ==> 0x20fc006069519369
/sys/class/fc_remote_ports/rport-3:0-1/port_state ==> Online
/sys/class/fc_remote_ports/rport-3:0-1/roles ==> Directory Server
/sys/class/fc_remote_ports/rport-3:0-1/scsi_target_id ==> -1
/sys/class/fc_remote_ports/rport-3:0-1/supported_classes ==> Class 2, Class 3
/sys/class/fc_remote_ports/rport-3:0-2/dev_loss_tmo ==> 35
/sys/class/fc_remote_ports/rport-3:0-2/maxframe_size ==> 2048 bytes
/sys/class/fc_remote_ports/rport-3:0-2/node_name ==> 0x2000000c5005dfae
/sys/class/fc_remote_ports/rport-3:0-2/port_id ==> 0x010ee1

```

```
/sys/class/fc_remote_ports/rport-3:0-2/port_name ==> 0x2100000c5005dfae
/sys/class/fc_remote_ports/rport-3:0-2/port_state ==> Online
/sys/class/fc_remote_ports/rport-3:0-2/roles ==> FCP Target
/sys/class/fc_remote_ports/rport-3:0-2/scsi_target_id ==> 0
/sys/class/fc_remote_ports/rport-3:0-2/supported_classes ==> Class 3
```

Information about targets visible to the HBA

```
/sys/class/fc_transport/target3:0:0/node_name ==> 0x2000000c5005dfae
/sys/class/fc_transport/target3:0:0/port_id ==> 0x010ee1
/sys/class/fc_transport/target3:0:0/port_name ==> 0x2100000c5005dfae
```

Information about SCSI devices visible to the HBA

```
/sys/class/scsi_device/3:0:0:0/device/delete ==> (File used for deleting a lun from SCSI database.)
/sys/class/scsi_device/3:0:0:0/device/device_blocked ==> 0
/sys/class/scsi_device/3:0:0:0/device/iocounterbits ==> 32
/sys/class/scsi_device/3:0:0:0/device/iDONE_cnt ==> 0x4a
/sys/class/scsi_device/3:0:0:0/device/ioerr_cnt ==> 0x1
/sys/class/scsi_device/3:0:0:0/device/iorequest_cnt ==> 0x4a
/sys/class/scsi_device/3:0:0:0/device/model ==> ST318453FC
/sys/class/scsi_device/3:0:0:0/device/queue_depth ==> 30
/sys/class/scsi_device/3:0:0:0/device/queue_type ==> simple
/sys/class/scsi_device/3:0:0:0/device/rescan ==> (File used for rescanning the storage)
/sys/class/scsi_device/3:0:0:0/device/rev ==> 0005
/sys/class/scsi_device/3:0:0:0/device/scsi_level ==> 4
/sys/class/scsi_device/3:0:0:0/device/state ==> running
/sys/class/scsi_device/3:0:0:0/device/timeout ==> 60
/sys/class/scsi_device/3:0:0:0/device/type ==> 0
/sys/class/scsi_device/3:0:0:0/device/vendor ==> SEAGATE
/sys/class/scsi_device/3:0:0:0/device/block:sdC/dev ==> 8:32
/sys/class/scsi_device/3:0:0:0/device/block:sdC/range ==> 16
/sys/class/scsi_device/3:0:0:0/device/block:sdC/removable ==> 0
/sys/class/scsi_device/3:0:0:0/device/block:sdC/size ==> 35843686
/sys/class/scsi_device/3:0:0:0/device/block:sdC/stat ==> 54 446 1004 136 0 0 0 0 104 136
/sys/class/scsi_device/3:0:0:0/device/block:sdC/queue/max_hw_sectors_kb ==> 32767
/sys/class/scsi_device/3:0:0:0/device/block:sdC/queue/max_sectors_kb ==> 512
/sys/class/scsi_device/3:0:0:0/device/block:sdC/queue/nr_requests ==> 128
/sys/class/scsi_device/3:0:0:0/device/block:sdC/queue/read_ahead_kb ==> 512
/sys/class/scsi_device/3:0:0:0/device/block:sdC/queue/scheduler ==> noop anticipatory deadline [cfq]
/sys/class/scsi_device/3:0:0:0/device/block:sdC/queue/iosched/back_seek_max ==> 16384
/sys/class/scsi_device/3:0:0:0/device/block:sdC/queue/iosched/back_seek_penalty ==> 2
/sys/class/scsi_device/3:0:0:0/device/block:sdC/queue/iosched/fifo_expire_async ==> 248
/sys/class/scsi_device/3:0:0:0/device/block:sdC/queue/iosched/fifo_expire_sync ==> 124
/sys/class/scsi_device/3:0:0:0/device/block:sdC/queue/iosched/quantum ==> 4
/sys/class/scsi_device/3:0:0:0/device/block:sdC/queue/iosched/queued ==> 8
/sys/class/scsi_device/3:0:0:0/device/block:sdC/queue/iosched/slice_async ==> 40
/sys/class/scsi_device/3:0:0:0/device/block:sdC/queue/iosched/slice_async_rq ==> 2
/sys/class/scsi_device/3:0:0:0/device/block:sdC/queue/iosched/slice_idle ==> 8
/sys/class/scsi_device/3:0:0:0/device/block:sdC/queue/iosched/slice_sync ==> 100
/sys/class/scsi_device/3:0:0:0/device/generic/dev ==> 21:3
/sys/class/scsi_device/3:0:0:0/device/scsi_generic:sg3 -> ../../../../../../class/scsi_generic/sg3
```

This document refers to various companies and products by their trade names. In most, if not all cases, their respective companies claim these designations as trademarks or registered trademarks. This information is provided for reference only. Although this information is believed to be accurate and reliable at the time of publication, Emulex assumes no responsibility for errors or omissions. Emulex reserves the right to make changes or corrections without notice. This report is the property of Emulex Corporation and may not be duplicated without permission from the Company.