



### FC HBA Driver for Linux Kernel 2.4.x

This software license applies only to QLogic customers.  
QLogic Corporation.  
All rights reserved.

### Table of Contents

1. [Package Contents](#)
2. [OS Support](#)
3. [Supported Features](#)
4. [Using the Driver Package](#)
  - 4.1 [Saving the Driver to Diskette](#)
  - 4.2 [Installing the Driver](#)
  - 4.3 [Using the SNIA API Library Package](#)
  - 4.4 [Enabling IP Support](#)
5. [Driver Parameters](#)
  - 5.1 [System Parameters](#)
  - 5.2 [NVRAM Parameters](#)
  - 5.3 [Driver Command Line Parameters](#)
6. [Additional Notes](#)
  - 6.1 [Proc Filesystem Support](#)
  - 6.2 [Failover Support](#)
  - 6.3 [Persistent Binding](#)
  - 6.4 [Configuration Data](#)
  - 6.5 [Forcing Device Rescan](#)
  - 6.6 [Booting from SAN](#)
  - 6.7 [Forcing a LIP](#)
  - 6.8 [Getting Firmware Dump](#)
7. [Contacting Support](#)

## 1. Package Contents

---

The following table describes the contents provided in the FC HBA Driver for Linux package.

Filename	Description
drvinstall	Linux installation script.
libinstall	Script file to install/setup HBA API library.
libremove	Linux script to remove libqlsdrm.so from /usr/lib or/and # # /usr/lib64.
qla2x00src-vx.yy.zz.tgz	Compressed package that contains the driver and all support documents. <b>NOTE:</b> <i>x.yy.zz</i> represents the driver package version number.
qlapi-vx.yybuildbb-rel.tgz	Compressed package that contains the HBA API software and all support documents. <b>NOTE:</b> <i>x.yy</i> represents the version and <i>bb</i> represents the build number.
readme.qla2x00	Text version of this package readme file.

## 2. OS Support

---

The FC HBA Driver for Linux is compatible with the following OS platforms.

Operating Systems		
OS Name	OS Type	Hardware Platform
Red Hat RHEL AS 3.0	32-bit/64-bit	Intel x86, Intel EM64T, AMD64, Intel IA64, and PPC64 platforms
Novell SLES 8	32-bit/64-bit	Intel x86, Intel AMD64, Intel IA64, and PPC64 platforms

**NOTE:** For specific OS service packs (SP) and updates, refer to the descriptions where this software version is posted on the QLogic website ([http://support.qlogic.com/support/drivers\\_software.asp](http://support.qlogic.com/support/drivers_software.asp)).

## 3. Supported Features

---

The FC HBA Driver for Linux supports the following:

- FCAL - direct attach loop
- Point-to-point
- Fabric support
- Initiator mode only
- Fault recovery on down loops
- Persistent binding
- Extended LUN support up to 255 LUNs
- FC tape support
- Non-Failover and Failover capability

## 4. Using the Driver

---

This section provides procedures for saving the driver to a diskette, installing the driver, and using the SNIA API library package.

## 4.1 Saving the Driver to Diskette

---

To save the driver distribution to a diskette:

1. Download the driver distribution file:
  - `qla2x00-vx.yy.zz-dist.tgz` or the driver source file
  - `qla2x00src-vx.yy.zz.tgz` from QLogic's website
2. If prompted "What would you like to do with this file?" choose **Save this file to disk**.
3. Insert a blank diskette and download to the diskette directly.

## 4.2 Installing the Driver

---

The source tgz package provides the content needed to build `qla2200.o` or `qla2300.o` for either UP or SMP systems:

- Build a single processor (UP) driver version (see [section 4.2.1](#)).
- Build a Symetric Multiprocess (SMP) driver version (see [section 4.2.2](#)).

After building the driver version, do one of the following:

- Manually load the driver using `insmod` or `modprobe` (see [section 4.2.3](#)).
- Create a ramdisk image to automatically load the driver (see [section 4.2.4](#)).

### 4.2.1 Building a Single Processor (UP) Driver Version

---

For **Red Hat RHEL Distributions**, enter the commands shown in the following steps:

1. Using the diskette created in [section 4.1](#), copy the distribution file (`qla2x00-vx.yy.zz-dist.tgz`) or the driver source file (`qla2x00src-vx.yy.zz.tgz`) to `/qla2x00`. Enter the following commands from the `/` (root) directory:

```
# mkdir qla2x00
# cd qla2x00
# mount /mnt/floppy
# cp /mnt/floppy/*.tgz . (the period at the end is required)
# tar -xvzf *.tgz
```

(Execute the following additional steps if you have copied the distribution file `qla2x00-vx.yy.zz-dist.tgz`)

```
# cd qla2x00
# ./drvinstall (this extracts the driver source files to the current directory)
```
2. Build the Driver `qla2200.o` and `qla2300.o` from the source code:

```
# make all
```

For **Novell SLES Distributions**, enter the commands shown in the following steps:

1. Install the kernel-source from the SLES distribution CD-ROM:

```
# yast -i kernel-source
```

or

```
# yast2 -i kernel-source
```
2. Create a soft link (`/usr/src/linux`) to the kernel source (`/usr/src/<linux-version>`):

```
# ln -sf /usr/src/<linux-version> /usr/src/linux
```
3. To ensure kernel version synchronization between the driver and running kernel (for kernel version below 2.4.21):

```
# cd /usr/src/linux
# make mrproper (completely clean the kernel tree)
# cp /boot/vmlinuz.config .config (copy the new config)
# make oldconfig (update configuration using .config)
# make dep (rebuild the dependencies)
# make modules (build the modules)
```

**NOTE:** The steps above do not apply to SLES kernel 2.4.21 and above since it is distributed with pre-build kernel-headers.

- Using the diskette you created in [section 4.1](#), to the root (/) directory and copy the distribution file (qla2x00-vx.yy.zz-dist.tgz) or the driver source file (qla2x00src-vx.yy.zz.tgz) to /qla2x00:
 

```
# mkdir qla2x00
# cd qla2x00
# mount /mnt/floppy
# cp /mnt/floppy/*.tgz . (the period at the end is required)
# tar -xvzf *.tgz
```

 (Run the following additional commands if you copied the qla2x00-vx.yy.zz-dist.tgz distribution file)
 

```
# cd qlagic
# ./drvinstall (this extracts the driver source files to the current directory)
```
- Build the Driver qla2200.o and qla2300.o from the source code:
 

For Kernel Version below 2.4.21:

```
# make all OSVER=linux
```

For Kernel Version 2.4.21 and above:

```
# make all OSVER=linux-include/<configuration-name>
```

For example, for kernel version 2.4.21-107-itanium2, itanium2 is the *configuration-name* and /usr/src/linux-include is a soft link to /usr/src/linux-2.4.21-107-include.

**NOTE:** To load the driver manually, see [section 4.2.3](#). To create a ramdisk image to load the driver during system boot time, see [section 4.2.4](#).

#### 4.2.2 Building a Symmetric Multiprocessor (SMP) Driver Version

For **Red Hat Distribution**, enter the commands shown in the following steps:

- Using the diskette you created in [section 4.1](#), go to the root (/) directory and copy the distribution file (qla2x00-vx.yy.zz-dist.tgz) or the driver source file (qla2x00src-vx.yy.zz.tgz) to /qla2x00:
 

```
# mkdir qla2x00
# cd qla2x00
# mount /mnt/floppy
# cp /mnt/floppy/*.tgz .
# tar -xvzf *.tgz
```

 (Run the following additional commands if you copied the qla2x00-vx.yy.zz-dist.tgz distribution file.)
 

```
# cd qlagic
# ./drvinstall (this extracts the driver source files to the current directory)
```
- Build the driver qla2200.o or qla2300.o from the source code:
 

```
# make all SMP=1
```

For **Novell SLES Distributions**, enter the commands shown in the following steps:

- Install the kernel-source from the SLES distribution CD-ROM:
 

```
# yast -i kernel-source or # yast2 -i kernel-source
```
- Create a soft link (/usr/src/linux) to the kernel source (/usr/src/<linux-version>):
 

```
# ln -sf /usr/src/<linux-version> /usr/src/linux
```
- To ensure kernel version synchronization between the driver and running the kernel (for kernel version below 2.4.21):
 

```
# cd /usr/src/linux
# make mrproper (completely clean the kernel tree)
# cp /boot/vmlinuz.config .config (copy the new .config)
# make oldconfig (update configuration using .config)
# make dep (rebuild the dependencies)
# make modules (build the modules)
```

**NOTE:** The steps above do not apply to SLES kernel 2.4.21 and above since it is distributed with pre build kernel headers.

4. Using the diskette you created in [section 4.1](#), go to the root (/) directory and copy the distribution file (qla2x00-vx.yy.zz-dist.tgz) or the driver source file (qla2x00src-vx.yy.zz.tgz) to /qla2x00:
 

```
# mkdir qla2x00
# cd qla2x00
# mount /mnt/floppy
# cp /mnt/floppy/*.tgz .
# tar -xvzf *.tgz
```

 (Run the following additional commands if you copied the qla2x00-vx.yy.zz-dist.tgz distribution file.)
 

```
# cd qllogic
# ./drvinstall (this extracts the driver source files to the current directory)
```
5. Build the driver qla2200.o or qla2300.o from the source code:
 

For Kernel Version below 2.4.21:

```
# make all SMP=1 OSVER=linux
```

For Kernel Version 2.4.21 and above:

```
# make all SMP=1 OSVER=linux-include/<configuration-name>
```

For example, for kernel version 2.4.21-107-itanium2, itanium2 is the configuration-name and /usr/src/linux-include is a soft link to /usr/src/linux-2.4.21-107-include.

**NOTE:** To load the driver manually, see [section 4.2.3](#). To create a ramdisk image to load the driver during system boot time, see [section 4.2.4](#).

### 4.2.3 Load the Driver Manually using insmod or modprobe

The driver module qla2300.o supports both 2Gb and 4Gb FC HBAs. For example, if a QLA234x HBA and a QLA246x HBA are installed in the same server, run the following command to initialize both HBAs:

```
# modprobe -v qla2300
```

Before loading the driver manually, build the driver binary from the driver source TGZ as described in sections [4.2.1](#) and [4.2.2](#).

To load the driver directly from the local build directory:

```
# insmod qla2200.o
```

or

```
# insmod qla2300.o
```

To load the driver using modprobe:

1. Build the driver qla2200.o or qla2300.o from the source code:
 

For **Red Hat Distribution**:

```
# make all install (build and copy driver to the right location)
```

For **SLES Distribution**:

```
# make all OSVER=linux install (build and copy driver to the right location)
```
2. Load the driver:
 

```
# modprobe -v qla2200
```

or

```
# modprobe -v qla2300
```

### 4.2.4 Create a ramdisk Image to Load the Driver

To create a ramdisk image:

1. Build the driver binary files (see [4.2.1](#) and [4.2.2](#)).
2. Install the driver:
 

For **Red Hat Distribution**:

```
# make all install
```

For **SLES Distribution**:

```
# make all OSVER=linux install
```

3. Add the following line to `/etc/modules.conf`:

For **Red Hat Distribution**:

```
alias scsi_hostadapter0 qla2200_conf
alias scsi_hostadapter1 qla2200
```

or

```
alias scsi_hostadapter0 qla2300_conf
alias scsi_hostadapter1 qla2300
```

**NOTE:** Add one entry for each HBA in the system. For example, if both QLA2200 and QLA2300 HBAs are installed add the following:

```
alias scsi_hostadapter0 qla2200_conf
alias scsi_hostadapter1 qla2200
alias scsi_hostadapter2 qla2300_conf
alias scsi_hostadapter3 qla2300
```

For **SLES Distribution**, modify the `/etc/sysconfig/kernel` file to specify which modules are added during ramdisk creation.

**NOTE:** Make sure the configuration module is listed before the actual driver module. For example:

```
INITRD_MODULES="aic7xxx qla2300_conf qla2300 qla2200_conf qla2200"
```

4. Enter the appropriate command.

For **Red Hat Distribution**:

```
# mkinitrd -f <ramdisk image file name> <kernel version>
```

**NOTE:** This command overwrites the original ramdisk image file if executed within the `/boot` directory. Specify a unique ramdisk image name to preserve the original ramdisk image.

- o On **IA-32**, copy the newly built `<ramdisk image file name>` file to `/boot`.
- o On **IA-64**, copy the newly built `<ramdisk image file name>` file to `/boot/efi/efi/RedHat`.

For **SLES Distribution**:

```
# /sbin/mk_initrd
```

By default, the system creates the ramdisk images as:

```
/boot/initrd
/boot/initrd.SLES
```

**NOTE:** This command overwrites the original ramdisk image file. To preserve the original ramdisk image specify a unique ramdisk image name:

```
# /sbin/mk_initrd -k <kernel name> -i <ramdisk image file name>
```

5. Configure the boot loader with the new ramdisk image for GRUB:

For **Red Hat Distribution**:

- a. Add `"initrd <ramdisk file name>"` in `/etc/grub.conf` file under one of the kernel entries to use the ramdisk image. For example:

```
kernel /vmlinuz-2.4.18-14 ro root=LABEL=/
initrd /initrd-2.4.18-14.img
```
- b. Reboot the system.

For **SLES Distribution**:

- a. Add `"initrd (hd0,1)/boot/<ramdisk file name>"` in `/boot/grub/menu.lst` file under one of the kernel entries to use the ramdisk image. For example:

```
kernel (hd0,1)/boot/vmlinuz root=/dev/hda2 max_scsi_luns=128
initrd (hd0,1)/boot/initrd
```
- b. Reboot the system.

### 4.3 Using the SNIA API Library Package

---

The SNIA API library package (`qlapi-<api_version>-rel.tgz`) is included in the driver combo package (`qla2x00-vx.yy.zz-dist.tgz`) or (`qla2x00-vx.yy.zz-fo-dist.tgz`).

Using the diskette you created in [section 4.1](#), go to the root (`/`) directory and copy the distribution file (`qla2x00-vx.yy.zz-dist.tgz`) to `/qla2x00`:

```
# mkdir qla2x00
# cd qla2x00
# mount /mnt/floppy
# cp /mnt/floppy/*.tgz . (the period at the end is required)
# tar -xvzf *.tgz
# cd qla2x00
```

- To install the SNIA API Library, enter the following command in the current directory to install/setup API library:  
# `./libinstall` (this installs/sets up HBA API library)
- To uninstall the SNIA API Library, enter the following command in the current directory to remove API library:  
# `./libremove` (script file to remove HBA API library)

### 4.4 Enabling IP Support

---

This section describes how to load the IP driver and configure the network interface to allow TCP/IP applications to communicate with IP-enabled FC HBAs. For details, see:

- [4.4.1 Loading the IP Driver](#)
- [4.4.2 Configuring Interfaces](#)

#### 4.4.1 Loading the IP Driver

---

Before loading the driver manually, build the driver binary from the driver source files as described in [section 4.2.2](#). Enter the `insmod` or `modprobe` commands to load the IP driver:

- To load the driver directly from the local build directory:  
# `insmod qla2300.o`  
# `insmod qla2xip.o`
- To load the driver using `modprobe`:
  - a. Install the driver module (`*.o`) files to the appropriate kernel module directory:  
# `make install`
  - b. Load the driver for `qla23xx` HBAs:  
# `modprobe -v qla2300`  
# `modprobe -v qla2xip`
- To unload the driver using `modprobe`:  
# `modprobe -r qla2xip`  
# `modprobe -r qla2300`

The `qla2xip` driver creates a network-interface binding to each IP-capable recognized HBA. After loading the IP driver, you can view the binding entries in the messages file. For example:

```
qla2xip: QLogic IP via Fibre Channel Network Driver
qla2xip: Driver Version 1.0b2, Entry point: e08e5060
qla2xip: Mapping interface fc0 to adapter 210100e08b20a15b
qla2xip: Mapping interface fc1 to adapter 210200e08b40a25b
```

## 4.4.2 Configuring Interfaces

---

To allow TCP/IP applications running on the host to communicate with other IP-capable FC HBAs, you must configure the network interfaces:

1. Compile the following basic host information to allow the IP driver to pass TCP/IP data over FC HBAs in your host:

```
Interface name: fc0          (From above)
    IP Address: 192.168.1.x*  (A non-routable address)
    Netmask: 255.255.255.0    (Standard class C mask)
Interface name: fc1          (From above)
    IP Address: 192.168.2.x*  (A non-routable address)
    Netmask: 255.255.255.0    (Standard class C mask)
```

\*x in the IP address is a unique number between (1 and 254)

2. Use the `ifconfig` program to configure an interface with the compiled host information:  
`# ifconfig fc0 192.168.1.x up`
3. Configure other interfaces (if multiple IP-capable HBAs are present in the host):  
`# ifconfig fc1 192.168.1.x up`
4. Verify the configured interfaces:  
`# ifconfig`

The screen displays details on the newly configured interfaces, as shown in the following example:

```
fc0 Link encap:Ethernet HWaddr 00:E0:8B:20:A1:5B
    inet addr:192.168.1.1 Bcast:192.168.1.255 Mask:255.255.255.0
    UP BROADCAST RUNNING MTU:4096 Metric:1
    RX packets:1214577458 errors:0 dropped:0 overruns:0 frame:0
    TX packets:1214213174 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:32
    RX bytes:3081095492 (2938.3 Mb) TX bytes:2751945609 (2624.4 Mb)

fc1 Link encap:Ethernet HWaddr 00:E0:8B:40:A2:5B
    inet addr:192.168.2.1 Bcast:192.168.2.255 Mask:255.255.255.0
    UP BROADCAST RUNNING MTU:4096 Metric:1
    RX packets:1204464697 errors:0 dropped:0 overruns:0 frame:0
    TX packets:1194873236 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:32
    RX bytes:1454694706 (1387.3 Mb) TX bytes:991094469 (945.1 Mb)
```

This completes configuration.

5. Verify that basic networking is possible between two hosts connected via two HBAs using a simple `ping` command:  
`# ping 192.168.1.2`

For more details on Linux networking, refer to the Linux Networking "HOWTOs" available at <http://www.tldp.org>.

## 5. Driver Parameters

---

This section describes the following driver parameters:

- [5.1 System Parameters](#)
- [5.2 NVRAM Parameters](#)
- [5.3 Driver Command Line Parameters](#)



## 5.1 System Parameters

---

You can configure support for multiple LUNs in one of three ways. Currently, the system can scan up to 255 LUNs that for each device (2.4.9-21 or above).

**NOTE:** If you have multiple HBAs, set `max_scsi_luns` to the largest number of LUNs supported by any one of these HBAs.

**If the SCSI Mid-Layer is compiled in the kernel, you can configure the boot loader to scan for multiple LUNs each time the system boots.**

*On IA-32, for GRUB, follow the steps for your OS platform.*

For **Red Hat Distribution:**

1. Append the `max_scsi_luns` parameters to each of the kernel images listed in the `/etc/grub.conf` file.

For example:

```
kernel /vmlinuz-2.4.7-10 ro root=/dev/hda2 max_scsi_luns=255
```

2. Reboot the system.

For **SLES Distribution:**

1. Append the `max_scsi_luns` parameters to each of the kernel images listed in the `/boot/grub/menu.lst` file. For example:

```
kernel (hd0,1)/boot/vmlinuz root=/dev/hda2 max_scsi_luns=25
```

2. Reboot the system.

*On IA-64, follow the steps for your OS platform.*

For **Red Hat Distribution:**

1. Add the following line to each of the kernel images listed in the `/boot/efi/efi/Red Hat/elilo.conf`.

For example:

```
append="max_scsi_luns=128"
```

2. Reboot the system.

For **SLES Distribution:**

1. Add the following line to each of the kernel images listed in the `/boot/efi/SLES/elilo.conf`. For example:

```
append="max_scsi_luns=128"
```

2. Reboot the system.

**If the SCSI Mid-Layer is compiled as a module:**

1. Add the following line to the `/etc/modules.conf` file to scan for multiple LUNs at each boot:

```
options scsi_mod max_scsi_luns=255
```

2. Rebuild the ramdisk image, as described in [section 4.2.3](#).

## 5.2 NVRAM Parameters

---

The driver reads the NVRAM parameters during initialization. Any changes made for the particular NVRAM feature in the *Fast!Util* take effect after reloading the driver or changing the parameters using the utilities in SANsurfer FC HBA Manager or SANsurfer FC CLI.

## 5.3 Driver Command Line Parameters

---

The driver gets its parameters from the command line or from `modprobe` option directive found in the `modules.conf` file. The parameters use the following format:

`<keyword>=value` format (for example, `ql2xfailover=1`)

where `<keyword>` is one of the option parameters described in the following paragraphs.

Usage: `modprobe -v qla2300 <keyword>=value`

Examples:

```
# modprobe -v qla2300.o ql2xfailover=1
```

```
# modprobe -v qla2300.o qlport_down_retry=60
```

### Command Line Parameter Descriptions

Parameter	Description	Default
ql2xfailover	This parameter defines whether failover mode is enabled or disabled. 0 to disable; 1 to enable.	0
ql2xmaxqdepth	This parameter defines the maximum queue depth reported to SCSI Mid-Level per device. The Queue depth specifies the number of outstanding requests per LUN.	32
ql2maxsgs	This parameter defines the maximum number of scatter gather entries reported to SCSI Mid-Level per request for the HBA.	32
ql2maxsectors	This parameter defines the maximum number of sectors reported to SCSI Mid-Level per request for the HBA.	512
ql2xlogintimeout	This parameter defines the Login timeout value in seconds during the initial login.	20 seconds
qlport_down_retry	This parameter defines how long to wait for a port that returns a PORT-DOWN status before returning I/O back to the OS.	0 (use value specified in NVRAM)
ql2xretrycount	This parameter defines the maximum number of SCSI Mid-Level retries allowed per command.	20 (standard mode value) 30 (failover mode value)
max_srbcs	This parameter defines the maximum number of simultaneous commands accepted from the SCSI Mid-Level per HBA.	4096
displayConfig	This parameter defines whether to display the current configuration. 1 - display the configuration 0 - don't display the configuration.	1
Binding method	This parameter defines what target persistent binding method to use: 0 - bind by Portname 1 - bind by PortID.	0 (portname binding)
extended_error_logging	This parameter defines whether to enable (1) or Disable (0) writing the debug information to /var/log/messages.	0 (disable)
MaxPathsPerDevice	This parameter defines maximum number of paths to a device.	8 (compile time only)
MaxRetriesPerPath	This parameter defines how many retries to perform on the current path before failing over to the next path in the path list.	3
MaxRetriesPerLo	This parameter defines total retries to do before failing the command and returning to the OS with selection timeout (DID_NO_CONNECT).	(MaxRetriesPerPath * MaxPathsPerDevice) + 1
QIFailoverNotifyType	This parameter defines type of failover notification mechanism to use when a failover or failback occurs. Certain storage systems require special CDBs issued to do failover or failback.	0 (none)
ConfigRequired	This parameter defines how to bind the devices. 0 - Present all the devices discovered to the OS. 1 - Present only the configured devices (i.e. the device defined in /etc/qla2300_conf) to the OS.	0
FailbackTime	This parameter defines the delay in seconds before performing a failback to ensure all paths are available.	5 seconds
RecoveryTime	This parameter defines the recovery time in seconds required before sending commands to the restored path.	10 seconds

ql2xprocessnotready	This parameter defines how the driver should handle a "NOT READY" indicator. 1 - Handled by the driver 0 - Disable the handling within the driver	1
ql2xautorestore	This parameter enables or disables the logic that restores the previous failed preferred path and/or controller for a given LUN. This option toggles the default state. Combine one or more of the following model numbers into an inclusion mask: 0x80 - MSA A/A (auto-restore disabled) 0x20 - HSV111, HSV101, HSV200, HSV210 (auto-restore disabled) 0x10 - DSXXX (auto-restore disabled) 0x04 - HSV110, HSV100 (auto-restore disabled) 0x02 - MSA1000 (auto-restore disabled) 0x01 - XP (auto-restore enabled)	0
ql2xlbType	This parameter defines the load Balance Method for the driver to use either static or dynamic: 0 (None) - Expose LUNs on the first active path and make them the preferred path or the first active optimize path and make them the preferred path (storages: MSA A/A and EVA A/A). 1 (Static load balancing) - Distribute and expose the LUNs across the active optimize ports or active un-optimize ports and HBAs. 2 (Least outstanding I/O) - Send command to the path with the lowest I/O count. 3 (Least Service time) - Send request to the path with the shortest execution time.	0
ql2xexcludemodel	This parameter excludes device models from being a failover capable target. Combine one or more of the following model numbers into an exclusion mask: 0x80 - MSA A/A 0x20 - HSV111, HSV101, HSV200, HSV210 0x10 - DSXXX, 0x04 - HSV110, HSV100 0x02 - MSA1000, 0x01 - XP	0
ql2xtgtemul	Enable/Disable target level grouping emulation. This option is necessary for the GUI to work correctly if the driver is set for LUN level grouping of paths by lunid. The following storages uses this method of combining paths: HSV210, DSXXX, HSV110, MSA1000, XP. 1 - Enable target level grouping emulation 0 - Disable target level grouping emulation	1 (Enable)

To view all available parameters, enter one of the following commands:

```
# modinfo -p qla2300
# modinfo -p qla2200
```

## 6. Additional Notes

---

This section provides the following additional information:

- [6.1 Proc Filesystem Support](#)
- [6.2 Failover Support](#)
- [6.3 Persistent Binding](#)
- [6.4 Configuration Data](#)
- [6.5 Forcing Device Rescan](#)
- [6.6 Booting from SAN](#)
- [6.7 Forcing a LIP](#)
- [6.8 Getting Firmware Dump](#)

### 6.1 Proc Filesystem Support

---

The `/proc` filesystem for the QLA2200, QLA23xx, and QLA246x driver are located in the `/proc/scsi/qla2200/` and `/proc/scsi/qla2300/` directories. These directories contain a file for each FC HBA in the system. Each file presents information about the HBA and transfers statistics for each discovered LUN.

### 6.2 Failover Support

---

This section describes how to provide failover support. For details, see the following topics:

- [6.2.1 How to Enable the Failover Support In The Driver](#)
- [6.2.2 Using SANsurfer FC HBA Manager with QLA2xxx Driver for the First Time](#)
- [6.2.3 Modifying the Configuration using SANsurfer FC HBA Manager \(GUI\)](#)

#### 6.2.1 How to Enable the Failover Support In The Driver

---

To implement failover support in the QLA2xxx driver, enable the macro `MPIO_SUPPORT` in `qla_settings.h` file. For example:

```
# define MPIO_SUPPORT 1
```

As an alternative, specify the option on the command line:

```
# modprobe -v qla2300 ql2xfailover=1
```

**NOTE:** The failover distribution package (`qla2x00-vx.yy.zz-fo-dist.tgz`) as this macro enabled by default.

#### 6.2.2 Using SANsurfer FC HBA Manager with QLA2xxx Driver for the First Time

---

To work with SANsurfer:

1. Build the driver with `MPIO_SUPPORT` set to 1 in `qla_settings.h`.
2. Add the following parameter in `modules.conf` file:  
options `qla2200 ql2xfailover=1`  
or  
options `qla2300 ql2xfailover=1`
3. Add as a command line parameter:  
# `modprobe -v qla2300 ql2xfailover=1`

### 6.2.3 Modifying the Configuration using SANsurfer FC HBA Manager (GUI)

The Linux SCSI mid-layer requires communicating with the target via LUN 0; therefore, the driver cannot mask LUN 0 when defined through the application. For the new LUN masking configuration to take effect, you must reload the driver.

To modify the configuration using the SANsurfer FC HBA Manager:

1. Load the driver:  
`modprobe <driver>`
2. Load the qlremote agent.
3. Start the GUI and connect it to the destination system.
4. Make LUN masking changes.
5. Disconnect the host from GUI and stop qlremote agent.
6. Unload the driver:  
`modprobe -r <driver>`
7. Reload the driver:  
`modprobe <driver>`
8. Load qlremote agent again.
9. Start the GUI and connect it to the destination system.

You should now see the updated LUN masking configuration.

**NOTE:** When using `modprobe` to load the driver, the length of the option line specified in `/etc/modules.conf` file has a limit of 2K characters. Any longer option line causes a string overflow error from `modprobe`.

### 6.3 Persistent Binding

The persistent binding information consists of some HBA configuration entries along with some target entries, which are specified in two formats: verbose and shorten. The shorten format allows a bigger configuration to fit in the limited space on the command line. As an alternate to using the command line, you can use the configuration file.

You can specify persistent binding in two ways: manually or using SANsurfer FC HBA Manager or SANsurfer FC HBA CLI. We recommend using SANsurfer tools for ease of use.

The following procedure shows how to manually add persistent binding commands.

The driver displays the current configuration when the `displayConfig` command line option is specified. The persistent binding configuration is found in `var/log/messages` file. It prints the configuration information in the format required by the driver. The best way to extract configuration messages is to use `grep` and direct the output to a file. You need to remove the Linux timestamp at the beginning each message and combine them together on single line.

For example:

```
# modprobe -v qla2300 displayConfig=1 # grep "scsi-qla" /var/log/messages > /tmp/info.cfg
```

The format of the persistent binding commands is as follows:

```
scsi-qla<#>-adapter-port=<HBA port name value>; OR  
<#>-w=<HBA port name value>; ** shorten format **
```

The designated by `qla<#>`, where the `<#>` is the HBA instance number.

The parameter specifies the FC port name to be used for the HBA where `<HBA port name value>` is the FC port name value in hexadecimal format. If this entry is not specified in the configuration file, the default value is the HBA's port name as saved in the NVRAM. For example:

```
scsi-qla00-adapter-port=210000e08b01158d\;
```

where host HBA instance 0 has a port name of 210000e08b01158d

```
scsi-qla<#1>-tgt-<#2>-di-<#3>-node=<device FC name>;
```

or

```
<#1>-<#2>-<#3>-n=<device FC name>; ** shorten format **
```



## 6.4.2 Compatibility with SANsurfer FC HBA Manager

---

QLA\_OPTS works seamlessly with updated SANsurfer FC HBA Manager applications. Originally, when the application saved a configuration, it saved the corresponding data to the `options` section of the `modules.conf` file in a form similar to the following:

```
ql2xopts=scsi-qla0-HBA-port=210000e08b000000\;scsi-qla0-tgt-1-di-0-node=20000020371682e7\;scsi-qla0-tgt-1-di-0-port=21000020371682e7\;scsi-qla0-tgt-1-di-0-pid=0000e2\;scsi-qla0-tgt-1-di-0-control=00\;
```

The application now saves the data to the appropriate `qla2[2|3]00.conf` file in `/etc` and then brands it to the binary file of the corresponding configuration module (`qla2200_conf.o` or `qla2300_conf.o`):

```
/lib/modules/<CURRENT_KERNEL_VERSION>/kernel/drivers/scsi
```

where `CURRENT_KERNEL_VERSION` is the result of the command `uname -r`.

This operation occurs automatically and requires no user intervention. However, if you loaded the driver from an `initrd` image, as before with the `modules.conf` interface, you would need to rebuild the `initrd` image after updating a configuration.

## 6.4.3 Updating the Configuration Manually

---

Manually updating the `/etc/qla2x00.conf` file is strongly discouraged. Because it can cause the binary module and the configuration file to get out of sync. If necessary, you can change the persistent/configuration manually.

To change a value manually:

1. Change the `/etc/qla2x00.conf` file.
2. Run `make install` from the driver's build directory to update the configuration module.

## 6.5 Forcing Device Rescan

---

Beginning with driver version `v6.06.00b12`, the driver provides a way to force the driver to re-scan the devices to allow you to add a new device. This triggers the driver to initiate LUN discovery process.

When scanning, first scan the driver (`qla2300/qla2200` driver etc), then the Linux scsi mid layer (such as an OS scan). For example, enter the commands:

```
# echo "scsi-qlascan" > /proc/scsi/<driver-name>/<HBA-id> (qlogic driver re-scans)
```

where

`<driver-name>` is `qla2100`, `qla2200`, or `qla2300`

`<HBA-id>` is the instance number of the HBA

After doing this, you can force the SCSI mid layer to perform its own scan and build the device table entry for the new device:

```
# echo "scsi add-single-device 0 1 2 3" >/proc/scsi/scsi  
(scsi mid layer re-scans)
```

where `0 1 2 3` is replaced by your Host Channel ID LUN.

## 6.6 Booting from SAN

---

To boot from SAN, create a driver disk (DD Kit) image, which enables you to install the Red Hat/SLES 8 OS and QLogic driver (qla2300) on the Fibre Channel boot disk connected to the HBA (QLA23XX/QLA63XX).

To build the driver disk image for **RH3 IA-32**, follow these steps:

1. Download `7.x_linux_driver_ddkit_for_Red_Hat_SLES_dist.tgz` from the QLogic website:  
<http://www.qlogic.com/>
2. Untar the `tgz` file:  

```
# tar -xvzf *.tgz
```
3. Enter:  

```
# dd if=qla2x00-vx.yy.zz-x86-dd-RHEL3-U5.img of=/dev/fd0 bs=1440k
```

  
(requires a 1.44MB floppy)

## 6.7 Forcing a LIP

---

The following NVRAM parameters must be set in order to perform the LIP reset:

- Enable Lip Reset
- Enable Target Reset (if the attached targets should be also be reset)

If both the above parameters are disabled, a Full Login LIP is executed.

Execute the following command to initiate the LIP reset process:

```
# echo "scsi-qlalip " > /proc/scsi/qla2300/<host_no>
```

## 6.8 Getting Firmware Dump

---

The `extras` directory in the driver source contains a utility used to retrieve the firmware dump on a specific host. To know if firmware dump has happened in the driver, search for "Firmware dump" in `/var/log/` messages.

For example:

```
# grep "Firmware dump" /var/log/messages
localhost kernel: scsi(0): Firmware dump saved to temp buffer (0/e0306000).
                    ^_ 0 is the host_no
```

To retrieve this firmware dump, enter the following command:

```
# cd extras
# ./qla_gdmp 0
Info: Firmware dumped to file fw_dump_0_20061128_152019.bin
Info: Firmware dump on 0 -- dumped 324586 bytes...
```

The utility saves the dump file in the current directory with a `.bin` extension.

**NOTE:** The dump can be retrieved only once.

The dump is in binary format, to convert the binary format into text format, enter the following command:

```
./qla_dmp < <FW dump bin file name>
```

For example:

```
# ./qla_dmp < fw_dump_0_20061128_152019.bin
```



## 7. Contacting Support

---

Please feel free to contact your QLogic approved reseller or QLogic Technical Support at any phase of integration for assistance. QLogic Technical Support can be reached by the following methods:

Web: <http://support.qlogic.com>

North America Contact Information

Email: [support@qlogic.com](mailto:support@qlogic.com)

Phone: (952) 932-4040

Support contact information for other regions of the world is available at the QLogic website:

<http://support.qlogic.com>

[Go to Top](#)



© Copyright 2007. All rights reserved worldwide. QLogic, the QLogic logo, and the Powered by QLogic logo are registered trademarks of QLogic Corporation. All other brand and product names are trademarks or registered trademarks of their respective owners.