QLOGIC

Readme File

FC HBA Driver for Mac OS X

Table of Contents

1. Package Contents

The FC HBA Mac OS X driver package is a compressed file (`QLogicHBA23xx-x.y.z.tgz`), which includes a tar file with the driver files needed to run on a Mac OS X platform. It also includes the following support documents:

- `readme.rtf` - readme file
- `release.txt` - release notes

2. OS Support

The FC HBA Driver for Mac OS X is compatible with the following OS platforms:

- MAC OS X Panther Version - 10.3.8 and above.
- MAC OS X Tiger Version - 10.4.8 and below.

3. Supported Features

The FC HBA Driver for Mac OS X supports the following:

- Persistent Binding
- FCAL - direct attach loop
- Point-to-point
- Fabric support
- Initiator mode only
- Fault recovery on down loops

4. Installing and Loading the Driver

The following subsections describe driver installation:

4.1 Installing from the Standard Package

The installer utility installs the driver in the `/System/Library/Extensions` folder with `root:wheel` permission. For details, see the following subsections:

4.1.1 Uncompress the Package

Uncompress the driver installation package `QLogicHBA23xx-x.y.z.tgz` as follows:

1. Open the terminal application, located at:
   `/Applications/Utilities/Terminal`
2. Run the following command to uncompress:
   `# tar -xvzf QLogicHBA23xx-x.y.z.tgz`

NOTEs:

- If the driver installation package is saved with a `.tar` extension, run the following command to uncompress the package:
  `#tar -xvf QLogicHBA23xx-x.y.z.tar`
- Uncompressing the file generates a `QLogicHBA23xx-x.y.z.pkg` installation package.
- If you uncompress the driver installation package using third-party utilities, the system creates a `QLogicHBA23xx-x.y.z` folder, which contains the installation package (`QLogicHBA23xx-x.y. z.pkg`).

4.1.2 Install the Driver

To install the FC HBA Driver for Mac OS X:

1. Open the Finder Application.
2. Go to the directory where the driver installation package (`QLogicHBA23xx-x.y.z.pkg`) is located.
3. Double-click the package icon and wait for the user interface screen to appear.
4. Proceed through the interactive installation GUI interface.
   NOTE: When running the installer with a non-root account, the installation program prompts for the Administrator password; enter the administrator password to install the driver.
5. Reboot the system for the newly installed driver to be loaded.

NOTEs:

- To load the driver manually without rebooting the system, refer to section 4.2 (Not Recommended).
- By default, the installer logs its messages in the `/var/log/system.log` file.

4.2 Loading the Driver Manually

The following subsections describe how to load and unload the driver manually:

- 4.2.1 Load the Driver Manually using kextload
- 4.2.2 Unload the Driver using kextunload

NOTEs:

- Before loading the driver manually, please follow section 4.1.
- Manual loading/unloading requires Administrative privileges.

4.2.1 Load the Driver Manually using kextload

To manually load the FC HBA Driver for Mac OS X:

1. Run the following command in the terminal application to load the driver:
   `# sudo kextload /System/Library/Extensions/QLogicHBA23xx.kext`
2. Enter Administrative password if prompted to continue loading.

4.2.2 Unload the Driver using kextunload

To unload the FC HBA Driver for Mac OS X:

1. Enter the following command in Terminal Application:
   `# sudo kextunload -b com.qlogic.driver.QLogicHBA23xx`
2. Enter Administrative password if prompted to continue unloading.

4.3 Preventing Driver from Automatically Loading at Boot Time

To prevent the FC HBA Mac OS X driver from automatically loading at boot time:

1. Move the installed driver from `/System/Library/Extensions` to some other folder. For example, enter the following command in the Terminal application:
   `# sudo mv /System/Library/Extensions/QLogicHBA23xx.kext <Destination Folder>`
2. Enter the administrative password if prompted.

5. Driver Parameters

The following subsections describe the Mac OS driver parameters:

- 5.1 NVRAM Parameters
- 5.2 Driver Parameters from Info.plist

5.1 NVRAM Parameters

The driver reads the NVRAM parameters at initialization time. You can dynamically change some of the NVRAM parameters using the SANsurfer HBA Manager or SANsurfer CLI application.

5.2 Configuring Driver Parameters from Info.plist

You can change the default value of the driver parameters in the `Info.plist` file, which is located in the `/System/Library/Extensions/QLogicHBA23xx.kext/Contents/` folder.

5.2.1 Driver Parameters

Parameter Option	Description	Default/Range
`osfailover`	This parameter enables/disables MAC OS failover support by driver is enable or disable.	Default: True False - Disable failover True - Enable failover
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.	Default: 0 (use value specified in NVRAM)
configRequired	This parameter defines how to bind the devices.	Default: 0 0=Present all the devices discovered to the OS 1=Present only the configured devices to the OS
Binding method	This parameter defines what type of target persistent binding method to use.	Default: 0 0=Bind by Portname 1=Bind by PortID.
ql2xlogintimeout	This parameter defines the Login timeout value in seconds during the initial login.	Default: 20 seconds
displayConfig	This parameter defines whether to display the current configuration.	Default: 1 1=Display the configuration 0=Don't display the configuration
max_srbs	This parameter defines the maximum number of simultaneous commands that can be accepted by HBA driver from SCSI subsystem.	Default: 1024
extended_error_logging	This parameter defines whether to enable (1) or disable (0) writing the debug information to /var/log/system.log.	Default: 0 (disable) 1=Enable writing the debug information 0=Disable writing the debug information
qfull_retry_count	This parameter defines how many times HBA driver will retry the received command with TASK_SET_FULL status.	Default: 24
qfull_retry_delay	This parameter defines the time period between 2 retries for a command when received back with TASK_SET_FULL status.	Default: 4 secs
qla2x00_reg_trace_enable	This parameter controls whether to display SCSI I/O statistics and driver debugging information in the IORegistery or not.	Default: true true=Enable display in IORegistery false=Disable display in IORegistery
qla2x00_verbose	This parameter defines whether to enable (true) or disable (false) writing the driver events information to the file /var/log/system.log during driver initialization time.	Default: true true=Enable writing driver events false=Disable writing driver events
qla2x00_reinit	This parameter defines whether to reinitialize the HBA hardware after the loop is down for more than 4 minutes.	Default: true true=Enable HBA hardware re-initialization false=disable initialization
qlogin_retry_count	This parameter defines the number of fabric logins that should be retried in case of a login failure.	Default: 0 (use value specified in NVRAM)
retry_gnnft	This parameter defines the number of get_node_name commands should be retried in case of failure.	Default: 10

5.2.2 Editing the Info.plist File

The OS keeps a cache of the driver's `Info.plist` file, which is required to update the OS cache when changing the driver parameters.

To edit the `Info.plist` file:

1. Open the `Info.plist` file using a login with Administrative permission.
2. Modify the parameter value in the file.
3. Save the file.
4. Update the system cache for `Info.plist` settings using following command in terminal application:
   `#sudo touch /System/Library/Extensions`
5. Reboot the system to make the changes take effect.

NOTE: Manual loading or unloading is not recommended. Reboot the system to make the changes take effect. If you decide to use the manual process, make sure that the driver has completed initialization.

6. Additional Notes

This section provides the following additional information:

6.1 IORegistry Support

You can view the driver parameters and HBA related information in the system IORegistry in one of the following two ways:

- Use the `IORegisteryExplorer` application located in the `/Developer/Applications/Utilities` folder.
- Type the following command in the terminal application:
  `# ioreg -l`

6.2 Failover Support

- MAC OS X has the failover support built-in the SAM family; it is enabled by default.
- MAC OS X Load Balancing and HBA side Failover are supported.
- At this time MAC OS X does not support Target Side Failover.
- MAC OS X failover support is prone to workloop dead lock, which sometimes causes I/Os to hang when HBAs are in multipathing configuration.

6.3 Persistent Binding

The Persistent Binding information consists of adapter configuration entries along with target entries. You can specify Persistent Binding in three ways:

- Manually
- Using SANsurfer HBA Manager
- Using SANsurfer CLI

NOTE: QLogic recommends using SANsurfer FC HBA Manager or SANSurfer CLI for ease of use.

The driver displays the current configuration in `/var/log/system.log` file, when the `displayConfig` option is set to `1`. The configuration information is in the format required by the driver. The best way to extract configuration messages is to use the `grep` command and direct the output to a file. You need to remove the MAC OS timestamp at the beginning of each message and combine them together on single line. For example:
`#grep "scsi-qla" /var/log/system.log> /tmp/info.cfg`

Persistent binding configuration format is as follows:

Device descriptions

scsi-qla<#>-adapter-port=<*adapter port name value*>;

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

scsi-qla0-adapter-port=210000e08b01158d\;

host adapter instance 0 has a portname of 210000e08b01158d

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

This parameter associates the specified <*device FC name*> with the SCSI target ID value specified by <*#2*> and a device ID value specified by <*#3*>, where <*device FC name*> type is the FC node name of the device, <*#2*> is the SCSI target ID to be assigned to the device, and <*#3*> is the device unique ID.

where

<*#1*> specifies the adapter instance number

<*#2*> specifies the SCSI ID of target

<*#3*> specifies the path/device ID

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

This parameter associates the specified <device FC name> with the SCSI target ID value specified by <*#2*> and a device ID value specified by <*#3*>, where <*device FC name*> type is the FC port.

where

<*#1*> specifies the adapter instance number

<*#2*> specifies the SCSI ID of Target

<*#3*> specifies the path/device ID (always 0 for non-failover) name of the device, <*#2*> is the SCSI target ID to be assigned to the device, and <*#3*> is the device unique ID.

scsi-qla<#1>-tgt-<#2>-di-<#3>-pid=<#4>;

When Bind setting is set to Bind By Port ID in Info.plist settings, this parameter associates the port ID specified by <#4> with target ID specified by <#2>.

where

<*#1*> specifies the adapter instance number

<*#2*> specifies the SCSI ID of target

<*#3*> specifies the path/device ID

<*#4*> specifies the port ID.

scsi-qla<#1>-tgt-<#2>-di-<#3>-disabled=<256 bit mask>;

This parameter associates the specified <256 bit mask> with the SCSI target ID value specified by <*#2*> and a device ID value specified by <*#3*>.

where

<*#1*> specifies the adapter instance number

<*#2*> specifies the SCSI ID of Target

<*#3*> specifies the path/device ID

<256 bit mask>

```
msb                                                                 lsb
0000000000000000000000000000000000000000000000000000000000000000F
```

The mask above makes the first four LUNs, 3, 2, 1, and 0 of a given target disabled on that target/path. This mask specification is heavily type checked to be a sequence of 64 hex digits.

6.4 Configuration Data

Configuration/persistent data is read from the configuration KEXT during driver loading time. Normally, configuration data is written via the SANsurfer FC HBA Manager or the scli application.

NOTEs:

- Approximately 300K of configuration space has been pre-allocated within the `QLogicHBA23xxConfig` KEXT for configuration/persistent data storage.
- The `qla_opts` utility works seamlessly with updated SANsurfer applications.
- The information is written to the appropriate `QLogicHBA23xxConfig.conf` file in `/etc` and then branded to the binary file of the corresponding configuration module (`QLogicHBA23xxConfig`): `/System/Library/Extensions/QLogicHBA23xxConfig.kext`

6.5 SNIA API Library Package

The current driver package doesn't contain the SNIA API Library Package.

6.6 Limitations

- Manually loading or unloading the driver using `kextload` or `kextunload` is not recommended.
- Manually unloading the driver during Active I/O is never recommended.
- Cable pulls on the Tape devices during Active I/O phase are not supported.

6.7 Known Issues

- This driver does not support Mac OS X on Intel-based CPUs.
- Target Device is not reported during the Name Server Query

Prior to QLogic Fibre Channel SANbox2 Switch Firmware Version (5.0.0.20.00), moving a target from one port to another switch port prevents the target device from being reported to the driver during Name Server query for the list of devices.

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
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