

Using the QLogic EFI Configuration and Diagnostic Protocols

Revision History	4
Introduction	5
Changing the display mode	5
EFI Driver Configuration	6
Starting the Configuration Protocol	6
Command summary.....	7
NVRAM Parameters	8
edit_adapter_settings	8
edit_advanced_settings	8
edit_database	8
edit_efi	9
Information.....	10
help.....	10
info.....	10
show_database	10
show_translation.....	10
show_buffer.....	10
Operation	11
abandon	11
write.....	11
quit.....	11
Detailed Command Descriptions	12
Detailed Description: edit_database	12
Detailed Description: edit_efi	12
Enable Alternate Boot Device	12
Enable Boot Order List	12
Enable Selective Login.....	13
Enable Selective Lun Logins	13
Set Variable EFIFCScanLevel	13
SAN Environment Features	14
WWN Database.....	14
Selective Login	14



Persistent Binding.....	14
Alternate Boot Device Selection	15
EFI Driver Diagnostics	16
Starting the Diagnostic Protocol	16
Standard Level Diagnostics.....	17
Extended Level Diagnostics	18
Manufacturing Level Diagnostics.....	19
Manufacturing Level Diagnostics Command Summary.....	21
connection.....	22
help.....	22
interface	22
iterations.....	22
loopback	22
mailbox	23
path.....	23
post.....	23
quit.....	23

Revision History

Date	Rev	Reason	By	Comments
9-6-02	1.0	Initial	T. Leonard	
10-7-02	1.1	Feature additions	T. Leonard	Added boot options to configuration protocol.
11-13-02	1.2	Feature additions	T. Leonard	Addes Scsi options, selective boot description.
03-21-03	1.3	Major Additions and revision	T. Leonard	
07-01-03	1.4	Added Boot Order List optins	T. Leonard	
09-25-03	1.5	Change requests	J. Carnuccio	
11-17-03	1.6	Manufacture mode update	J. Carnuccio	Clarified manufacture mode with examples.
11-18-03	1.7	Added post command	J. Carnuccio	
11-19-03	1.8	Minor changes	d. Wagner	Manju changes

Introduction

The QLogic efi driver configuration protocol provides display and modification of configuration parameters stored in the QLA23xx adapter nvram. The QLogic efi driver diagnostic protocol provides a minimal set of adapter diagnostics. Both protocols are provided as part of the QLogic efi scsi pass-thru protocol driver. Either protocol may be invoked from the efi shell. Each protocol uses a Command Line Interface (CLI) to allow the user to display and modify adapter configuration and diagnostic parameters; batch mode is not provided.

Changing the display mode

To make reading the efi shell information easier, change the number of lines displayed to the maximum allowed on your machine; first, at the efi shell enter the following command to see a list of available row and column dimension pairs:

```
mode
```

For example, the largest dimensioned pair may be listed as:

```
col 80 row 50
```

To select this mode, enter the following command:

```
mode 80 50
```

EFI Driver Configuration

During the configuration process, all changes are made to a local copy of the nvram; the changes may be committed to the adapter nvram by using the `write` command; prior to using this command, changes to the local copy may be abandoned by using the `abandon` or `quit` commands.

Starting the Configuration Protocol

For a more detailed description of the efi commands used in this procedure, refer to the efi shell documentation provided with your computer or the efi project documentation at:

<http://www.intel.com/technology/efi/index.htm>

The Configuration protocol is started from the efi shell using the following procedure:

First, enter the following command:

```
drivers
```

A list of installed efi drivers is displayed in table form; under the heading **DRIVER NAME**, find the driver with name **HP 2Gb Fibre Channel Driver**, and take note of its driver handle number in the **DRV** column (this number will be referred to as *driverhandle*).

There may be one or more driver instances listed; if only one driver instance is listed, then it is managing one or more QLA23xx adapters; if more than one driver instance is listed, then each instance is managing a subset of the available QLA23xx adapters.

Second, enter the following command:

```
drvcfg driverhandle
```

A list of adapters managed by this driver is displayed; take note of the controller handle number inside the brackets labeled **Ctrl[]** (this number will be referred to as *controllerhandle*).

Third, enter the following command to start the configuration protocol:

```
drvcfg -s driverhandle controllerhandle
```

A version banner is displayed, followed by the following prompt indicating `eficfg` mode:

```
eficfg>
```

A brief description of each `eficfg` command is displayed by entering the `help` command.

Command summary

Below is listed the three categories of commands, and under each category is listed the commands of that category:

NVRAM Parameters:

- edit_adapter_settings
- edit_advanced_settings
- edit_database
- edit_efi

Information:

- help
- info
- show_database
- show_translation
- show_buffer

Operation:

- abandon
- write
- quit

NVRAM Parameters

edit_adapter_settings

Used to display and modify the adapter settings; produces the following sequence:

```
Adapter Settings:
  Enable Hard Loop Id
  Hard Loop Id
  Loop Reset delay
  Enable Fibre Channel Tape Support
Frame Size:
  512
  1024
  2048
Connection Option:
  0 - Loop Only
  1 - Point To Point Only
  2 - Loop Preferred, Otherwise Point-to-Point
Data Rate:
  0 - 1 Gb/s
  1 - 2 Gb/s
  2 - Auto
```

edit_advanced_settings

Used to display and modify the advanced settings; produces the following sequence:

```
Advanced Adapter Settings:
  Operation Mode:
    0 - Interrupt for every I/O completion
    5 - Interrupt when Interrupt Delay Timer expires
    6 - Interrupt when Interrupt Delay Timer expires or no activity
  Interrupt Delay Timer(100 ms)
  Execution Throttle
  Login Retry Count
  Port Down Retry Count
  Link Down Timeout(seconds)
  LUNs per Target
  Enable Extended Error Logging
  Enable LIP Reset
  Enable LIP Full Login
  Enable Target Reset
```

edit_database

Used to display and modify entries in the wwn (world wide name) database.

Refer to **Detailed Description: edit_database** below.

edit_efi

Used to display and modify the efi parameters; produces the following sequence:

EFI parameters:

```
Enable Alternate Boot Device
Enable Boot Order List (disables Selective Login)
Enable Selective Logins
Enable Selective LUN Logins
Set Variable EFIFCScanLevel
```

Refer to **Detailed Description: edit_efi** below.

Information

help

Displays a brief list of available commands.

info

Displays the following adapter information:

1. adapter efi device path
2. adapter port wwn (wwpn)
3. adapter serial number
4. adapter ssvi d and ssdi d from nvram

The device path may be used to determine which adapter configuration is being displayed.

show_database

Displays the contents of the wwn database in table form.

show_translation

Displays the scsi target id translation table; this table is a list of scsi tid and fibre channel loop id mapping pairs; each entry in the table consists of the following for each device:

```
scsi id (tid)
fc loop id (lid)
world wide port name (wwpn)
world wide node name (wwnn)
```

All numbers are in hexadecimal; tid values from 0x00 to 0x0A are persistent, and tid values above 0x0A are assigned sequentially as devices are discovered; lid values above 0x80 are fabric attached while those below 0x7f are arbitrated loop attached.

show_buffer

Displays the contents of the local nvram buffer in hexadecimal; this is the local buffer containing changes made prior to using the **write** command to commit them to the adapter nvram.

Operation

abandon

Abandons the changes in the current configuration protocol local buffer and reloads its contents from the adapter nvram.

write

Writes the current configuration protocol local buffer to the adapter nvram; it may be used any time a command has been used to modify configuration data; when the nvram has been successfully written, the configuration protocol local buffer is reloaded from the adapter nvram.

quit

Quits the configuration and returns to the efi shell; the **write** command must be used to write any changes to the adapter nvram prior to quitting.

Detailed Command Descriptions

Detailed Description: `edit_database`

Displays and modifies a particular entry in the wwn database; the first parameter is the number of the entry in the database to be displayed; this is an integer between 0 and 8; the second parameter is the port wwn (wwpn) for the selected entry; this is a 16 digit hex number with the current value displayed in parenthesis; if database entries 0 or 1 are selected, a third parameter, the node wwn (wwnn) is displayed; this is a 16 digit hex number also; entering 0 for the node wwn causes the EFI SPT driver to treat this database entry like a port wwn only entry.

Detailed Description: `edit_efi`

The following boot options are listed by `edit_efi` and are described in detail below:

Enable Alternate Boot Device

Enable Boot Order List

Enable Selective Login

Enable Selective Lun Logins

Set Variable EFIFCScanLevel

Enable Alternate Boot Device

Specifies that the driver is to use wwn database entries 0 and 1 as the primary and alternate boot devices; if this option is enabled, the device with wwpn and wwnn in database entry 1 will be reported to efi as the primary boot device if the device with wwpn and wwnn in database entry 0 is not found; if the primary boot device is found, the secondary boot device is not reported to efi; when this option is not selected, entries 0 and 1 are both reported if found.

Enable Boot Order List

Disables both **Selective Login** and **Selective Lun Logins**; the driver determines which devices to login to based on one of two methods: the first method retrieves a list of devices from the san, and each device wwn is compared to the controller wwn database and matching devices are logged in; the second method, enabled by this option, searches the boot order list for entries that contain the adapter path followed by a fibre channel device path node, and any matching entries are saved in a internal list, and then the controller retrieves a list of devices from the san and any devices matching the internal list are logged in and reported to efi; this method is used to boot from those operating systems that create boot order lists containing hardware device path entries such as HPUX; operating systems such as Windows and Linux create device path entries that do not contain information about the location of the boot device and thus this method can not be used.

Enable Selective Login

Specifies that the driver is to use the wwn database as a list of devices that the adapter is permitted to login; enable this option to limit the adapter device discovery to devices matching those in the wwn database.

Enable Selective Lun Logins

Used with `Selective Login` which logs in only those devices in the adapter wwn database; if this option is not enabled, all luns present on any logged in device will be presented to efi; if this option is enabled, only the lun specified in the wwn database associated with the device will be logged into; this eliminates scanning and presenting a large number of luns to efi when only one is desired.

Set Variable EFIFCScanLevel

Used with `Boot Order List Login`; the variable `EFIFCScanLevel` is maintained by efi in the system nvram; if the variable is not defined or set to 0, then only devices in the `Boot Order List` will be logged in and reported to efi; if this variable is set to any non zero value, then all devices found on the san will be logged in and reported to efi; the configuration protocol allows this variable to be created if it does not exist; if the variable does not exist, the menu will display a message asking if it should be created.

SAN Environment Features

There are two methods of controlling the adapter login of asn devices; the first method, **Selective Login**, uses a list of device wwns and luns created and maintained in the adapter wwn database; the second method, **Boot Order List Login**, searches the boot order list for device paths matching the adapters that contain fibre channel device path nodes; the choices address the two different methods by which OS loaders place boot device paths in the efi boot order list.

The **Selective Login** method also provides **Selective Lun Logins**, persistent binding, and **Alternate Boot Device** selection; all these features use the adapter wwn database.

WWN Database

The efi driver configuration protocol is used to create and maintain the wwn database for each adapter. Each adapter reserves the first 6 scsi target ids (tids) for use with persistent binding; the default tid for the adapter is 0; the next 5 tids are bound to the entries in the wwn database; database entry 0 corresponds to tid 1, database entry 2 corresponds to tid 2, and so on.

All data base entries have a port wwn field; entry 0 has a node wwn field and an 8 bit lun field; entry 1 has a node wwn, and it uses the lun field of entry 0; entries 2 to 4 have 16 bit lun fields.

Configuration command **show_database** displays the wwn database, and command **edit_database** is used to display and modify any particular database entry.

Selective Login

Selective Login allows the system administrator to select which fabric devices the adapter will login to and report to the system. When this feature is enabled, up to 5 devices may be specified; at driver binding start time, the driver queries the fabric simple name server (sns) for devices; each wwn received from the sns is compared against the entries in the wwn database.; the adapter will only login those wwns that are found in the database; this limits the number of devices that are logged in. **Selective LUN Login** is used to select one lun on each device that was selected using the wwn database; when **Selective LUN Login** is disabled, all luns on all logged in devices are reported to efi; enabling **Selective LUN Login** limits the number of luns per device reported to one. So, at most 5 device/lun combinations will be reported to efi.

Persistent Binding

Persistent binding allows the system administrator to bind a device to a particular scsi id; to use this feature, the system administrator places the wwpn of the device to be bound in the wwn database of the adapter; at driver binding start time, the driver searches for devices on both the fabric and loop; each wwpn that is found is compared to the wwn database and any match assigns to the device the scsi id corresponding to the entry in the database.

Alternate Boot Device Selection

Alternate Boot Device selection allows the system administrator to define and bind a primary and an alternate device to a single scsi id; both the port and node wwns of each device may be specified; at driver binding start time, the driver looks for the primary device; if it is found, the device is bound to a fixed scsi id and the alternate device is ignored; if the primary device is not found, the driver searches for the alternate device, and if found, the alternate device is bound to the same fixed scsi id as the primary device.

EFI Driver Diagnostics

Starting the Diagnostic Protocol

For a more detailed description of the efi commands used in this procedure, refer to the efi shell documentation provided with your computer or the EFI project documentation at:

<http://www.intel.com/technology/efi/index.htm>.

There are three levels of diagnostics available:

- Standard
- Extended
- Manufacturing

The selection of the diagnostic level is made using a command line switch.

The Diagnostic protocol is started from the efi shell using the following procedure:

First, enter the following command:

```
drivers
```

A list of installed efi drivers is displayed in table form; under the heading **DRIVER NAME**, find the driver with name **QLcFC SCSI PASS Thru Driver**, and take note of its driver handle number in the **DRV** column (this number will be referred to as *driverhandle*).

There may be one or more driver instances listed; if only one driver instance is listed, then it is managing one or more QLA23xx adapters; if more than one driver instance is listed, then each instance is managing a subset of the available QLA23xx adapters.

Second, enter the following command:

```
drvdiag driverhandle
```

A list of adapters managed by this driver is displayed; take note of the controller handle number inside the brackets labeled **Ctrl[]** (this number will be referred to as *controllerhandle*).

Third, refer to the sections below describing the level of diagnostics required.

Standard Level Diagnostics

Enter the following command to start the standard interactive diagnostic:

```
drvdiag -s driverhandle controllerhandle
```

This performs a mailbox test on the adapter registers followed by a loopback test at the 10-bit interface; at the end of the test, a disconnect/connect sequence is done to restart the adapter.

Example:

```
fs0:\> drvdiag -s 6b 26
```

```
Run Diagnostics
```

```
Drv[6B] Ctrl[26] Lang[eng] Mailbox Test:
```

```
Mailbox test passed.
```

```
FC Loop Back Test: 1 iteration Loopback point: 10 bit interface
```

```
Loopback Test Passed
```

```
Available Devices:
```

```
Entry: 0 Target: 0 Lun: 00000000 LoopId: 82
```

```
Entry: 1 Target: 0 Lun: 00000001 LoopId: 82
```

```
Entry: 2 Target: 1 Lun: 00000000 LoopId: 81
```

```
- PASSED
```

Extended Level Diagnostics

Enter the following command to start the extended interactive diagnostic:

```
drvdiag -e driverhandle controllerhandle
```

This performs a mailbox test on the adapter registers followed by loopback tests at both the 10-bit interface and the 1-bit interface; and then runs an SRAM memory test. At the end of the test, a disconnect/connect sequence is done to restart the adapter.

Example:

```
fs0:\> drvdiag -e 6b 26
Run Diagnostics
Drv[6B] Ctrl[26] Lang[eng] Mailbox Test:
Mailbox test passed.
FC Loop Back Test: 10 iterations Loopback point: 10 bit interface
Loopback Test Passed
FC Loop Back Test: 10 iterations Loopback point: 1 bit interface
Loopback Test Passed
Memory Test
Sizing SRAM...
Testing Address 00010000
Testing Address 00018000
Testing Address 00020000
RISC SRAM size: 131072 words
Starting Test:
Pattern Result
0000 PASS
FFFF PASS
5555 PASS
AAAA PASS
Address PASS
Available Devices:
Entry: 0 Target: 0 Lun: 00000000 LoopId: 82
Entry: 1 Target: 0 Lun: 00000001 LoopId: 82
Entry: 2 Target: 1 Lun: 00000000 LoopId: 81
- PASSED
```

Manufacturing Level Diagnostics

Enter the following command to start the manufacturing interactive diagnostic protocol:

```
drvdiag -m driverhandle controllerhandle
```

A version banner is displayed, followed by the following prompt indicating `efidiag` mode:

```
efidiag>
```

A brief description of each `efidiag` command is displayed by entering the `help` command.

Example:

```
fs0:\> drvdiag -e 6b 26
```

```
Run Diagnostics
```

```
Drv[6B] Ctrl[26] Lang[eng] Adapter Path: Acpi(PNP0A03,00)/Pci(01|01)
```

```
efidiag> path
```

```
Adapter Path: Acpi(PNP0A03,00)/Pci(01|01)
```

```
efidiag> connection
```

```
FC Connection State: UP
```

```
FC Connection Type: FPort
```

```
efidiag> interface
```

```
Available loopback interafce:
```

```
0 - 10 bit loopback
```

```
1 - 1 bit loopback
```

```
2 - external loopback
```

```
Loopback interface [0]? 1
```

```
efidiag> iterations
```

```
Iteration count [1]? 10
```

```
efidiag> loopback
```

```
FC Loop Back Test: 10 iterations Loopback point: 1 bit interface
```

```
Loopback Test Passed
```

```
efidiag> mailbox
```

```
Mailbox Test:
```

```
Mailbox test passed.
```

```
efidiag> quit
```

```
Exiting...
```

```
- PASSED
```

Manufacturing Level Diagnostics Command Summary

The manufacturing level interactive commands are listed below:

- connection
- help
- interface
- iterations
- loopback
- mailbox
- path
- post
- quit

connection

Displays the state of the adapter's fibre channel connection; there are two states:

UP
DOWN

If the connection is UP, then the type of connection is displayed:

LOOP
POINT-POINT

help

Displays a brief list of available commands.

interface

Display and sets the interface for the loopback test; the selections are:

10-bit
1-bit
external

iterations

Displays and sets the iteration count for the loopback test; the available range is 1 to 10000.

loopback

Executes the loopback test; the loopback test builds and transmits a known 4096 byte pattern, and compares the received pattern with the transmitted pattern; there are several loopback test failure reasons:

Loopdown

External loopback point was selected but the loop connection is down.

Failure due to loop errors

Displays the crc error count and the loopback point where the failure occurred.

Data miscompare

Received data does not match transmitted data.

mailbox

Executes the mailbox register test on the adapter's ISP23xx chip.

path

Displays the adapter efi path; used to verify that diagnostics are running on the correct adapter.

post

Execute the POST diagnostic test on the current adapter.

quit

Quits diagnostic mode and returns to the efi shell.