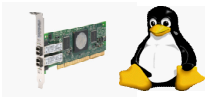
WHITE PAPER

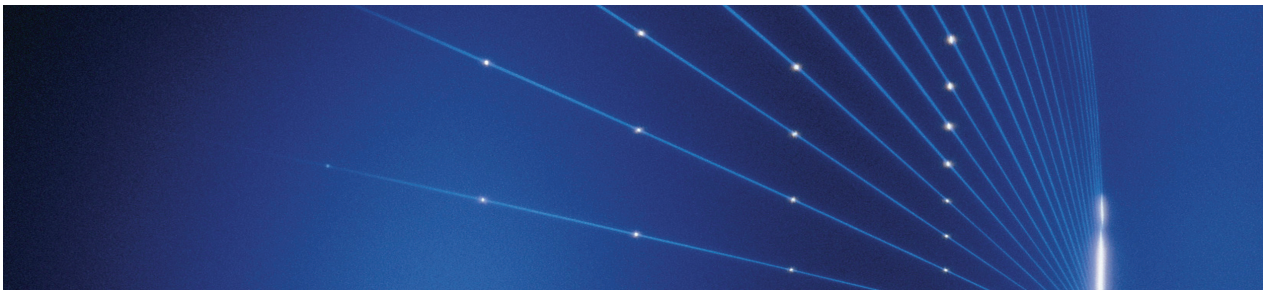
QLOGIC

# Persistent Naming Using *udev* in Linux Environments



## QLogic Linux Fibre Channel (FC) Solutions

## *Enable Ease of Storage Device Management with Intuitive and Persistent Device File Names*



### Executive Summary

QLogic has been delivering industry leading Fibre Channel (FC) solutions in Linux since 1999. QLogic offers a wide range of management tools for Linux that ease the burden on Storage Area Network (SAN) administrators. These tools include SANsurfer FC HBA Manager with GUI and CLI interfaces, diagnostic tools, and utilities. In addition, the QLogic FC Linux driver *qla2xxx* includes comprehensive support of new features (such as *udev*) included in the Linux 2.6 kernel.

This white paper provides a detailed set of instructions that allow SAN administrators to use *udev* functionality to build a persistent naming scheme for storage devices (LUNs) connected to QLogic FC Host Bus Adapters (HBAs). Using *udev* along with a few simple rules, a SAN administrator can enable applications to access LUNs using consistent and intuitive device file names.

### Key Findings

- QLogic FC drivers are at the forefront of leveraging enterprise Linux features in delivering ease-of-use to data center administrators.
- The new Linux 2.6 kernel features leveraged by the QLogic Linux Driver are:
  - *sysfs* enables user-space access to device and driver attributes.
  - Using *sysfs* functionality, *udev* enables user-space device file naming based on user-defined rules.
- The QLogic FC driver *qla2xxx* complies with the new driver programming model in the Linux 2.6 kernel and supports *sysfs* functionality, thereby enabling simple *udev* rules that allow SAN administrators to organize device file names intuitively and persistently.
- In large SAN configurations, the *udev* mechanism eases storage device management by enabling intuitive and persistent naming of device file names. The mechanism seamlessly extends to boot devices as well.

## Overview

In Linux, storage devices are represented by a device file. For example, the device file `/dev/hda` represents the first IDE disk discovered on the host; the device file `/dev/sda` represents the first SCSI drive discovered on the host, etc.

Traditionally, all devices are listed under the `/dev` directory. However, as storage needs in enterprise data centers grow, the number of devices under the `/dev` directory increase. For a SAN administrator, it is easier to manage a large storage configuration if the storage infrastructure in Linux supports the ability to:

- Easily associate a device file with the storage device it controls
- Maintain an association between the device file and the storage device persistently across reboots
- List `/dev` entries only for those devices that are accessible to the host

In Linux kernel 2.6, *udev* addresses this functionality in conjunction with the *sysfs* functionality. Each of these features is described in the following paragraphs. For additional information, see the References listed in this document.

- ***sysfs***: The *sysfs* file system is an in-memory, virtual file system that provides a view of the kernel internal data structure (*kobject*) hierarchy. Using *sysfs*, SAN administrators can view the device topology of their system as a simple file system. The *sysfs* file system is similar to the traditional proc file system; however, the *sysfs* file system shows the hierarchical relationships among the components of the device driver model. *sysfs* builds the topology using driver attributes and variables from kernel internal data structures. A SAN administrator can view and manipulate driver attributes using the *sysfs* interface.
- ***udev***: *udev* is the new way of managing the `/dev` directory. It is run from user space. *udev* creates a dynamic `/dev` directory and provides consistent device naming across reboots (when configured as such), and provides a user space Application Programming Interface (API) to access information about current system devices. *udev* relies on device information available through *sysfs*.

The related top-level directories of this file system are described in the following paragraphs. These directories allow the SAN administer to easily map a device with the associated device file name.

- **block/**. All block devices, independent of the bus to which they are connected
- **devices/**. All hardware devices recognized by the kernel, organized according to the bus to which they are connected
- **bus/**. The buses in the system that host the devices
- **drivers/**. The device drivers registered with the kernel
- **class/**. The types of devices in the system; the same class may include devices hosted by different buses and driven by different drivers
- **power/**. Files to handle the power states of some hardware devices
- **firmware/**. Files to handle the firmware of some hardware devices

QLogic's FC driver *qla2xxx* has implemented support for *sysfs* in the Linux 2.6 kernel. The following sections provide specific instructions for using *udev* to build persistent naming for storage devices connected through QLogic FC HBAs. This white paper also contains instructions for setting up to boot from non-default device names, as created through *udev* persistent naming.

## How to Use *udev*

To create and name `/dev` device files corresponding to devices that are present in the system, *udev* relies on matching the information provided by *sysfs* with the rules provided by users. Files listing the *udev* rules are contained in the `/etc/udev/rules.d` directory. Using the *udev* rules files, a user can specify the naming conventions for any device for which the user wants a name that is different from the one that the kernel assigns by default.

### *udev* `.rules` Files

In SuSE Linux Enterprise Server™ 10, for example, the following `.rules` files can be found under the `/etc/udev/rules.d` directory:

```
root:/dev/disk# ls /etc/udev/rules.d
05-udev-early.rules          60-persistent-storage.rules
29-net_trigger_firmware.rules 64-device-mapper.rules
30-net_persistent_names.rules65-cdrom.rules
31-network.rules             71-multipath.rules
40-alsa.rules                72-multipath-compat.rules
50-udev-default.rules        80-sysconfig.rules
56-iedma.rules               85-mount-fstab.rules
60-persistent-in put.rules   90-hal.rules
                             95-udev-late.rules
```

Among these files, the `60-persistent-storage.rules` file lists rules for persistent device binding for storage devices. This file creates symbolic links to default kernel names (i.e. device files `hd*` and `sd*`).

The files under `/dev/disk` are symbolic links to default device files created by the kernel. QLogic recommends keeping the default device files that kernel assigns, as there are other agents in the kernel that use the traditional default names. Consequently, a

symbolic link to the default name is preferred instead of overriding the default naming convention.

### Persistent Naming of Storage Devices Connected through QLogic FC HBA

SCSI storage devices, both disks and tapes, are identified by the Universally Unique ID (UUID). The following procedure (using *udev* rules) allows a user to identify the storage devices connected through QLogic FC HBAs and build persistent device file name mapping to these devices:

1. Find the device files starting with `sd` under the /dev directory.
2. Validate that the storage device is connected to an HBA that is marked with "BUS" as "PCI".
3. Validate that the device is connected to a QLogic FC HBA by comparing the "driver" attribute from the *sysfs* file system with the QLogic FC driver *qla2xxx*.
4. Get the vital product data UUID from the device by issuing a SCSI Inquiry command with page code 83h.
5. Create a symbolic link to the matched device file; QLogic recommends including a keyword, such as `qlgc-fc-lun`, to easily identify the device.

The following paragraphs contain a sample list of entries in the *udev* .rule file for this procedure:

```
# persistent storage links: /dev/{disk,tape}/{by-
id,by-uuid,by-label,by-path,by-name}
# scheme based on "Linux persistent device names",
2004, Hannes Reinecke <hare@suse.de>
...

# by-id (hardware serial number)
# QLogic

KERNEL=="sd*[!0-9]", ENV{ID_SERIAL}=="?*",
IMPORT{program}="/sbin/scsi_id -p 0x83 -g -s /block/
%k", SYSFS{driver}=="qla2xxx", SYMLINK+="disk/by-id/
qlgc-fc-$env{ID_BUS}-$env{ID_SERIAL}"

KERNEL=="sd*[0-9]", ENV{ID_SERIAL}=="?*", SYSFS{drive
r}=="qla2xxx", SYMLINK+="disk/by-id/qlgc-fc-$env{ID_
BUS}-$env{ID_SERIAL}-part%n"

# by-path (shortest physical path)
# QLogic

KERNEL=="sd*[!0-9]", ENV{ID_SERIAL}=="?*",
IMPORT{program}="/sbin/scsi_id -p 0x83 -g -s /block/
%k", SYSFS{driver}=="qla2xxx", SYMLINK+="disk/by-
path/qlgc-fc-$env{ID_PATH}"

KERNEL=="sd*[0-9]", ENV{ID_SERIAL}=="?*", SYSFS{dr
iver}=="qla2xxx", SYMLINK+="disk/by-path/qlgc-fc-
$env{ID_PATH}-part%n"
```

The first example looks for raw devices; the second example looks for any partitions in each of the raw devices. The above listed rules use `/sbin/scsi_id`, which is included as part of the *udev* package.

These rules can be added to the `60-persistent-storage.rules` file. A separate `.rule` file is not recommended. A management application or a configuration utility can identify and find the persistent device file by comparing the sysfs attribute of each entry in the `.rules` file listed under the `/etc/udev/rules.d` directory.

Files in the `/etc/udev/rules.d` directory are parsed in lexical order. In some cases, the order in which the rules are parsed is important. In general, QLogic-specific rules (if needed) must be parsed before the defaults. This parsing can be done by creating a file name lexically lower than the `60-persistent-storage.rules` file. For example, a rules file named `59-qlogic-persistent-storage.rules` is processed before the default rules file.

Following the rules under the `60-persistent-storage.rules` file, *udev* creates symbolic links to the device files under the three sub-directories:

- `./by-id`
- `./by-path`
- `./by-uuid`

## Example Setup with Dell PowerEdge, EMC CLARiiON CX3, and QLogic QLA2462

This section provides an example of persistent binding for the following configuration:

- Dell® PowerEdge® 2850 system running SLES 10
- EMC® CLARiiON® CX3 with 15 LUNs configured
- QLogic QLA2462 4-Gb PCI-X FC HBA

The following text is the list of device files created under the `/dev/disk` directory. These files remain persistent over reboots.

**Listing under /dev/disk:**

```
total 0
drwxr-xr-x  6 root root  120 Jul 17 11:59 .
drwxr-xr-x 10 root root 7120 Jul 17 12:00 ..
drwxr-xr-x  2 root root  780 Jul 17 12:00 by-id
drwxr-xr-x  2 root root   80 Jul 17 12:00 by-label
drwxr-xr-x  2 root root  660 Jul 17 12:00 by-path
drwxr-xr-x  2 root root  220 Jul 17 12:00 by-uuid
```

**Listing under /dev/disk/by-id:**

```
total 0
drwxr-xr-x 2 root root 400 2007-02-22 21:27 .
drwxr-xr-x 5 root root 100 2007-02-22 21:26 ..
lrwxrwxrwx 1 root root   9 Jul 17 12:00 edd-int13_dev80 -> ../../sda
lrwxrwxrwx 1 root root  10 Jul 17 12:00 edd-int13_dev80-part1 -> ../../sda1
lrwxrwxrwx 1 root root  10 Jul 17 12:00 edd-int13_dev80-part2 -> ../../sda2
lrwxrwxrwx 1 root root  10 Jul 17 12:00 edd-int13_dev80-part3 -> ../../sda3
lrwxrwxrwx 1 root root  10 Jul 17 12:00 edd-int13_dev80-part4 -> ../../sda4
lrwxrwxrwx 1 root root   9 Jul 17 12:00 edd-int13_dev81 -> ../../sdb
lrwxrwxrwx 1 root root  10 Jul 17 12:00 edd-int13_dev81-part1 -> ../../sdb1
lrwxrwxrwx 1 root root   9 Jul 17 12:00 qlgc-fc-scsi-320000004cf0333ca -> ../../sde
lrwxrwxrwx 1 root root  10 Jul 17 12:00 qlgc-fc-scsi-320000004cf0333ca-part1 -> ../../sde1
lrwxrwxrwx 1 root root   9 Jul 17 12:00 qlgc-fc-scsi-320000004cf43ab48 -> ../../sdf
lrwxrwxrwx 1 root root  10 Jul 17 12:00 qlgc-fc-scsi-320000004cf43ab48-part1 -> ../../sdf1
lrwxrwxrwx 1 root root   9 Jul 17 12:00 qlgc-fc-scsi-320000004cf75e849 -> ../../sdc
lrwxrwxrwx 1 root root  10 Jul 17 12:00 qlgc-fc-scsi-320000004cf75e849-part1 -> ../../sdc1
lrwxrwxrwx 1 root root   9 Jul 17 12:00 qlgc-fc-scsi-320000004cfa88ccb -> ../../sdd
lrwxrwxrwx 1 root root  10 Jul 17 12:00 qlgc-fc-scsi-320000004cfa88ccb-part1 -> ../../sdd1
lrwxrwxrwx 1 root root   9 Jul 17 11:59 scsi-SATA_WDC_WD2000JS-00_WD-WCANKA779269 -> ../../sdb
lrwxrwxrwx 1 root root  10 Jul 17 11:59 scsi-SATA_WDC_WD2000JS-00_WD-WCANKA779269-part1 -> ../../sdb1
lrwxrwxrwx 1 root root   9 Jul 17 11:59 scsi-SATA_WDC_WD800JD-00L_WD-WMAM9FP91708 -> ../../sda
lrwxrwxrwx 1 root root  10 Jul 17 11:59 scsi-SATA_WDC_WD800JD-00L_WD-WMAM9FP91708-part1 -> ../../sda1
lrwxrwxrwx 1 root root  10 Jul 17 11:59 scsi-SATA_WDC_WD800JD-00L_WD-WMAM9FP91708-part2 -> ../../sda2
lrwxrwxrwx 1 root root  10 Jul 17 11:59 scsi-SATA_WDC_WD800JD-00L_WD-WMAM9FP91708-part3 -> ../../sda3
lrwxrwxrwx 1 root root  10 Jul 17 11:59 scsi-SATA_WDC_WD800JD-00L_WD-WMAM9FP91708-part4 -> ../../sda4
```

**Listing under /dev/disk/by-path:**

```
total 0
drwxr-xr-x 2 root root 420 2007-02-22 21:27 .
drwxr-xr-x 5 root root 100 2007-02-22 21:26 ..
lrwxrwxrwx 1 root root   9 Jul 17 12:00 pci-0000:00:1f.1-ide-0:0 -> ../../hda
lrwxrwxrwx 1 root root   9 Jul 17 12:00 pci-0000:00:1f.2-scsi-0:0:0:0 -> ../../sda
lrwxrwxrwx 1 root root  10 Jul 17 12:00 pci-0000:00:1f.2-scsi-0:0:0:0-part1 -> ../../sda1
lrwxrwxrwx 1 root root  10 Jul 17 12:00 pci-0000:00:1f.2-scsi-0:0:0:0-part2 -> ../../sda2
lrwxrwxrwx 1 root root  10 Jul 17 12:00 pci-0000:00:1f.2-scsi-0:0:0:0-part3 -> ../../sda3
lrwxrwxrwx 1 root root  10 Jul 17 12:00 pci-0000:00:1f.2-scsi-0:0:0:0-part4 -> ../../sda4
lrwxrwxrwx 1 root root   9 Jul 17 12:00 pci-0000:00:1f.2-scsi-1:0:0:0 -> ../../sdb
lrwxrwxrwx 1 root root  10 Jul 17 12:00 pci-0000:00:1f.2-scsi-1:0:0:0-part1 -> ../../sdb1
lrwxrwxrwx 1 root root   9 Jul 17 12:00 qlgc-fc-pci-0000:09:00.1-fc-0x22000004cf0333ca:0x0000000000000000
-> ../../sde
lrwxrwxrwx 1 root root  10 Jul 17 12:00 qlgc-fc-pci-0000:09:00.1-fc-0x22000004cf0333ca:0x0000000000000000-
part1 -> ../../sde1
lrwxrwxrwx 1 root root   9 Jul 17 12:00 qlgc-fc-pci-0000:09:00.1-fc-0x22000004cf43ab48:0x0000000000000000
-> ../../sdf
```

```
lrwxrwxrwx 1 root root  10 Jul 17 12:00 qlgc-fc-pci-0000:09:00.1-fc-0x22000004cf43ab48:0x0000000000000000-
part1 -> ../../sdf1
lrwxrwxrwx 1 root root   9 Jul 17 12:00 qlgc-fc-pci-0000:09:00.1-fc-0x22000004cf75e849:0x0000000000000000
-> ../../sdc
lrwxrwxrwx 1 root root  10 Jul 17 12:00 qlgc-fc-pci-0000:09:00.1-fc-0x22000004cf75e849:0x0000000000000000-
part1 -> ../../sdc1
lrwxrwxrwx 1 root root   9 Jul 17 12:00 qlgc-fc-pci-0000:09:00.1-fc-0x22000004cfa88ccb:0x0000000000000000
-> ../../sdd
lrwxrwxrwx 1 root root  10 Jul 17 12:00 qlgc-fc-pci-0000:09:00.1-fc-0x22000004cfa88ccb:0x0000000000000000-
part1 -> ../../sdd1

Listing under /dev/disk/by-uuid:
total 0
drwxr-xr-x 2 root root  60 2007-02-22 21:26 .
drwxr-xr-x 5 root root 100 2007-02-22 21:26 ..
lrwxrwxrwx 1 root root  10 2007-02-22 21:26 237daae6-f0ed-4e0e-a59d-839d4d48cadd -> ../../sda2
```

### Persistent Binding of Storage Devices at Boot Time

The following prerequisites must to be met to use persistent binding at boot time:

- Boot from a *mkinitcpio initramfs* image
- Enable *udev* in the `/etc/mkinitcpio.conf` file.
- When the *initramfs* image is generated, version 101-3 or greater of `klibc-udev` must be installed (persistent naming does not work in earlier versions).

When updating `klibc-udev` from an earlier version and persistent naming is used, generate the *initramfs* image before rebooting.

In the following example, `/dev/sda1` is the root partition. The SAN administrator wants to call it `boot-disk` for persistent naming. In the grub `menu.lst` file, the kernel line using the default device file name looks like this:

```
kernel /boot/vmlinuz26 root=/dev/sda1 vga=0x318 ro
```

Depending on the naming scheme used, the kernel line can be changed to the following:

```
kernel /boot/vmlinuz26 root=/dev/disk/by-label/
book-disk vga=0x318 ro
```

## Summary and Conclusion

Persistent device naming provides a unique and sticky device file to each physical device present on the system. Using the udev feature and the rules provided in this white paper, a SAN administrator can:

- Maintain persistent mapping between device file names to LUNs connected and QLogic FC HBAs
- Assign intuitive device file names to ease the burden of managing LUNs in large SAN configurations
- Use intuitive and persistent names to boot devices on the SAN
- Provide consistent device file names to enterprise applications without causing any disruptions to operating system applications that may rely on default device file names assigned by the kernel

As shown in this white paper, *udev* rules to manage LUNs connected through QLogic FC HBAs are simple to construct and use. QLogic actively contributes to the Linux community efforts to expand Fibre Channel functionality in Linux and simplify SAN wide storage management.

## References

1. Linux Device Naming - by Matt Domsch and Ahmad Ali
http://www.dell.com/downloads/global/power/ps1q07-20060392-Domsch.pdf
2. Proceedings of the Linux Symposium 2003
http://www.kroah.com/linux/talks/ols_2003_udev_paper/Reprint-Kroah-Hartman-OLS2003.pdf
3. Writing *udev* rules
http://www.reactivated.net/writing_udev_rules.html