

README.DAC960.txt

Linux Driver for Mylex DAC960/AccelerRAID/eXtremeRAID PCI RAID Controllers

Version 2.2.11 for Linux 2.2.19

Version 2.4.11 for Linux 2.4.12

PRODUCTION RELEASE

11 October 2001

Leonard N. Zubkoff
Dandelion Digital
lnz@dandelion.com

Copyright 1998-2001 by Leonard N. Zubkoff <lnz@dandelion.com>

INTRODUCTION

Mylex, Inc. designs and manufactures a variety of high performance PCI RAID controllers. Mylex Corporation is located at 34551 Ardenwood Blvd., Fremont, California 94555, USA and can be reached at 510.796.6100 or on the World Wide Web at <http://www.mylex.com>. Mylex Technical Support can be reached by electronic mail at mylexsup@us.ibm.com, by voice at 510.608.2400, or by FAX at 510.745.7715. Contact information for offices in Europe and Japan is available on their Web site.

The latest information on Linux support for DAC960 PCI RAID Controllers, as well as the most recent release of this driver, will always be available from my Linux Home Page at URL "<http://www.dandelion.com/Linux/>". The Linux DAC960 driver supports all current Mylex PCI RAID controllers including the new eXtremeRAID 2000/3000 and AccelerRAID 352/170/160 models which have an entirely new firmware interface from the older eXtremeRAID 1100, AccelerRAID 150/200/250, and DAC960PJ/PG/PU/PD/PL. See below for a complete controller list as well as minimum firmware version requirements. For simplicity, in most places this documentation refers to DAC960 generically rather than explicitly listing all the supported models.

Driver bug reports should be sent via electronic mail to "lnz@dandelion.com". Please include with the bug report the complete configuration messages reported by the driver at startup, along with any subsequent system messages relevant to the controller's operation, and a detailed description of your system's hardware configuration. Driver bugs are actually quite rare; if you encounter problems with disks being marked offline, for example, please contact Mylex Technical Support as the problem is related to the hardware configuration rather than the Linux driver.

Please consult the RAID controller documentation for detailed information regarding installation and configuration of the controllers. This document primarily provides information specific to the Linux support.

DRIVER FEATURES

The DAC960 RAID controllers are supported solely as high performance RAID controllers, not as interfaces to arbitrary SCSI devices. The Linux DAC960 driver operates at the block device level, the same level as the SCSI and IDE

README.DAC960.txt

drivers. Unlike other RAID controllers currently supported on Linux, the DAC960 driver is not dependent on the SCSI subsystem, and hence avoids all the complexity and unnecessary code that would be associated with an implementation as a SCSI driver. The DAC960 driver is designed for as high a performance as possible with no compromises or extra code for compatibility with lower performance devices. The DAC960 driver includes extensive error logging and online configuration management capabilities. Except for initial configuration of the controller and adding new disk drives, most everything can be handled from Linux while the system is operational.

The DAC960 driver is architected to support up to 8 controllers per system. Each DAC960 parallel SCSI controller can support up to 15 disk drives per channel, for a maximum of 60 drives on a four channel controller; the fibre channel eXtremeRAID 3000 controller supports up to 125 disk drives per loop for a total of 250 drives. The drives installed on a controller are divided into one or more "Drive Groups", and then each Drive Group is subdivided further into 1 to 32 "Logical Drives". Each Logical Drive has a specific RAID Level and caching policy associated with it, and it appears to Linux as a single block device. Logical Drives are further subdivided into up to 7 partitions through the normal Linux and PC disk partitioning schemes. Logical Drives are also known as "System Drives", and Drive Groups are also called "Packs". Both terms are in use in the Mylex documentation; I have chosen to standardize on the more generic "Logical Drive" and "Drive Group".

DAC960 RAID disk devices are named in the style of the obsolete Device File System (DEVFS). The device corresponding to Logical Drive D on Controller C is referred to as /dev/rd/cCdD, and the partitions are called /dev/rd/cCdDp1 through /dev/rd/cCdDp7. For example, partition 3 of Logical Drive 5 on Controller 2 is referred to as /dev/rd/c2d5p3. Note that unlike with SCSI disks the device names will not change in the event of a disk drive failure. The DAC960 driver is assigned major numbers 48 - 55 with one major number per controller. The 8 bits of minor number are divided into 5 bits for the Logical Drive and 3 bits for the partition.

SUPPORTED DAC960/AccelerRAID/eXtremeRAID PCI RAID CONTROLLERS

The following list comprises the supported DAC960, AccelerRAID, and eXtremeRAID PCI RAID Controllers as of the date of this document. It is recommended that anyone purchasing a Mylex PCI RAID Controller not in the following table contact the author beforehand to verify that it is or will be supported.

eXtremeRAID 3000

- 1 Wide Ultra-2/LVD SCSI channel
- 2 External Fibre FC-AL channels
- 233MHz StrongARM SA 110 Processor
- 64 Bit 33MHz PCI (backward compatible with 32 Bit PCI slots)
- 32MB/64MB ECC SDRAM Memory

eXtremeRAID 2000

- 4 Wide Ultra-160 LVD SCSI channels
- 233MHz StrongARM SA 110 Processor
- 64 Bit 33MHz PCI (backward compatible with 32 Bit PCI slots)
- 32MB/64MB ECC SDRAM Memory

AccelerRAID 352

README.DAC960.txt

2 Wide Ultra-160 LVD SCSI channels
100MHz Intel i960RN RISC Processor
64 Bit 33MHz PCI (backward compatible with 32 Bit PCI slots)
32MB/64MB ECC SDRAM Memory

AcceleRAID 170

1 Wide Ultra-160 LVD SCSI channel
100MHz Intel i960RM RISC Processor
16MB/32MB/64MB ECC SDRAM Memory

AcceleRAID 160 (AcceleRAID 170LP)

1 Wide Ultra-160 LVD SCSI channel
100MHz Intel i960RS RISC Processor
Built in 16M ECC SDRAM Memory
PCI Low Profile Form Factor - fit for 2U height

eXtremeRAID 1100 (DAC1164P)

3 Wide Ultra-2/LVD SCSI channels
233MHz StrongARM SA 110 Processor
64 Bit 33MHz PCI (backward compatible with 32 Bit PCI slots)
16MB/32MB/64MB Parity SDRAM Memory with Battery Backup

AcceleRAID 250 (DAC960PTL1)

Uses onboard Symbios SCSI chips on certain motherboards
Also includes one onboard Wide Ultra-2/LVD SCSI Channel
66MHz Intel i960RD RISC Processor
4MB/8MB/16MB/32MB/64MB/128MB ECC EDO Memory

AcceleRAID 200 (DAC960PTL0)

Uses onboard Symbios SCSI chips on certain motherboards
Includes no onboard SCSI Channels
66MHz Intel i960RD RISC Processor
4MB/8MB/16MB/32MB/64MB/128MB ECC EDO Memory

AcceleRAID 150 (DAC960PRL)

Uses onboard Symbios SCSI chips on certain motherboards
Also includes one onboard Wide Ultra-2/LVD SCSI Channel
33MHz Intel i960RP RISC Processor
4MB Parity EDO Memory

DAC960PJ

1/2/3 Wide Ultra SCSI-3 Channels
66MHz Intel i960RD RISC Processor
4MB/8MB/16MB/32MB/64MB/128MB ECC EDO Memory

DAC960PG

1/2/3 Wide Ultra SCSI-3 Channels
33MHz Intel i960RP RISC Processor
4MB/8MB ECC EDO Memory

DAC960PU

1/2/3 Wide Ultra SCSI-3 Channels
Intel i960CF RISC Processor
4MB/8MB EDRAM or 2MB/4MB/8MB/16MB/32MB DRAM Memory

DAC960PD

1/2/3 Wide Fast SCSI-2 Channels
Intel i960CF RISC Processor
4MB/8MB EDRAM or 2MB/4MB/8MB/16MB/32MB DRAM Memory

README.DAC960.txt

DAC960PL 1/2/3 Wide Fast SCSI-2 Channels
 Intel i960 RISC Processor
 2MB/4MB/8MB/16MB/32MB DRAM Memory

DAC960P 1/2/3 Wide Fast SCSI-2 Channels
 Intel i960 RISC Processor
 2MB/4MB/8MB/16MB/32MB DRAM Memory

For the eXtremeRAID 2000/3000 and AcceleRAID 352/170/160, firmware version 6.00-01 or above is required.

For the eXtremeRAID 1100, firmware version 5.06-0-52 or above is required.

For the AcceleRAID 250, 200, and 150, firmware version 4.06-0-57 or above is required.

For the DAC960PJ and DAC960PG, firmware version 4.06-0-00 or above is required.

For the DAC960PU, DAC960PD, DAC960PL, and DAC960P, either firmware version 3.51-0-04 or above is required (for dual Flash ROM controllers), or firmware version 2.73-0-00 or above is required (for single Flash ROM controllers)

Please note that not all SCSI disk drives are suitable for use with DAC960 controllers, and only particular firmware versions of any given model may actually function correctly. Similarly, not all motherboards have a BIOS that properly initializes the AcceleRAID 250, AcceleRAID 200, AcceleRAID 150, DAC960PJ, and DAC960PG because the Intel i960RD/RP is a multi-function device. If in doubt, contact Mylex RAID Technical Support (mylexsup@us.ibm.com) to verify compatibility. Mylex makes available a hard disk compatibility list at <http://www.mylex.com/support/hdcomp/hd-lists.html>.

DRIVER INSTALLATION

This distribution was prepared for Linux kernel version 2.2.19 or 2.4.12.

To install the DAC960 RAID driver, you may use the following commands, replacing "/usr/src" with wherever you keep your Linux kernel source tree:

```
cd /usr/src
tar -xvzf DAC960-2.2.11.tar.gz (or DAC960-2.4.11.tar.gz)
mv README.DAC960 linux/Documentation
mv DAC960.[ch] linux/drivers/block
patch -p0 < DAC960.patch (if DAC960.patch is included)
cd linux
make config
make bzImage (or zImage)
```

Then install "arch/i386/boot/bzImage" or "arch/i386/boot/zImage" as your standard kernel, run lilo if appropriate, and reboot.

To create the necessary devices in /dev, the "make_rd" script included in "DAC960-Utilities.tar.gz" from <http://www.dandelion.com/Linux/> may be used. LILO 21 and FDISK v2.9 include DAC960 support; also included in this archive are patches to LILO 20 and FDISK v2.8 that add DAC960 support, along with statically linked executables of LILO and FDISK. This modified version of LILO

README.DAC960.txt

will allow booting from a DAC960 controller and/or mounting the root file system from a DAC960.

Red Hat Linux 6.0 and SuSE Linux 6.1 include support for Mylex PCI RAID controllers. Installing directly onto a DAC960 may be problematic from other Linux distributions until their installation utilities are updated.

INSTALLATION NOTES

Before installing Linux or adding DAC960 logical drives to an existing Linux system, the controller must first be configured to provide one or more logical drives using the BIOS Configuration Utility or DACCF. Please note that since there are only at most 6 usable partitions on each logical drive, systems requiring more partitions should subdivide a drive group into multiple logical drives, each of which can have up to 6 usable partitions. Also, note that with large disk arrays it is advisable to enable the 8GB BIOS Geometry (255/63) rather than accepting the default 2GB BIOS Geometry (128/32); failing to do so will cause the logical drive geometry to have more than 65535 cylinders which will make it impossible for FDISK to be used properly. The 8GB BIOS Geometry can be enabled by configuring the DAC960 BIOS, which is accessible via Alt-M during the BIOS initialization sequence.

For maximum performance and the most efficient E2FSCK performance, it is recommended that EXT2 file systems be built with a 4KB block size and 16 block stride to match the DAC960 controller's 64KB default stripe size. The command "mke2fs -b 4096 -R stride=16 <device>" is appropriate. Unless there will be a large number of small files on the file systems, it is also beneficial to add the "-i 16384" option to increase the bytes per inode parameter thereby reducing the file system metadata. Finally, on systems that will only be run with Linux 2.2 or later kernels it is beneficial to enable sparse superblocks with the "-s 1" option.

DAC960 ANNOUNCEMENTS MAILING LIST

The DAC960 Announcements Mailing List provides a forum for informing Linux users of new driver releases and other announcements regarding Linux support for DAC960 PCI RAID Controllers. To join the mailing list, send a message to "dac960-announce-request@dandelion.com" with the line "subscribe" in the message body.

CONTROLLER CONFIGURATION AND STATUS MONITORING

The DAC960 RAID controllers running firmware 4.06 or above include a Background Initialization facility so that system downtime is minimized both for initial installation and subsequent configuration of additional storage. The BIOS Configuration Utility (accessible via Alt-R during the BIOS initialization sequence) is used to quickly configure the controller, and then the logical drives that have been created are available for immediate use even while they are still being initialized by the controller. The primary need for online configuration and status monitoring is then to avoid system downtime when disk drives fail and must be replaced. Mylex's online monitoring and configuration utilities are being ported to Linux and will become available at some point in the future. Note that with a SAF-TE (SCSI Accessed Fault-Tolerant Enclosure)

README.DAC960.txt

enclosure, the controller is able to rebuild failed drives automatically as soon as a drive replacement is made available.

The primary interfaces for controller configuration and status monitoring are special files created in the /proc/rd/... hierarchy along with the normal system console logging mechanism. Whenever the system is operating, the DAC960 driver queries each controller for status information every 10 seconds, and checks for additional conditions every 60 seconds. The initial status of each controller is always available for controller N in /proc/rd/cN/initial_status, and the current status as of the last status monitoring query is available in /proc/rd/cN/current_status. In addition, status changes are also logged by the driver to the system console and will appear in the log files maintained by syslog. The progress of asynchronous rebuild or consistency check operations is also available in /proc/rd/cN/current_status, and progress messages are logged to the system console at most every 60 seconds.

Starting with the 2.2.3/2.0.3 versions of the driver, the status information available in /proc/rd/cN/initial_status and /proc/rd/cN/current_status has been augmented to include the vendor, model, revision, and serial number (if available) for each physical device found connected to the controller:

***** DAC960 RAID Driver Version 2.2.3 of 19 August 1999 *****

Copyright 1998-1999 by Leonard N. Zubkoff <lnz@dandelion.com>

Configuring Mylex DAC960PRL PCI RAID Controller

Firmware Version: 4.07-0-07, Channels: 1, Memory Size: 16MB

PCI Bus: 1, Device: 4, Function: 1, I/O Address: Unassigned

PCI Address: 0xFE300000 mapped at 0xA0800000, IRQ Channel: 21

Controller Queue Depth: 128, Maximum Blocks per Command: 128

Driver Queue Depth: 127, Maximum Scatter/Gather Segments: 33

Stripe Size: 64KB, Segment Size: 8KB, BIOS Geometry: 255/63

SAF-TE Enclosure Management Enabled

Physical Devices:

| | | | |
|-----|-------------|----------------|----------------|
| 0:0 | Vendor: IBM | Model: DRVS09D | Revision: 0270 |
|-----|-------------|----------------|----------------|

| | |
|----------------|------------|
| Serial Number: | 68016775HA |
|----------------|------------|

| | |
|--------------|-------------------------|
| Disk Status: | Online, 17928192 blocks |
|--------------|-------------------------|

| | | | |
|-----|-------------|----------------|----------------|
| 0:1 | Vendor: IBM | Model: DRVS09D | Revision: 0270 |
|-----|-------------|----------------|----------------|

| | |
|----------------|------------|
| Serial Number: | 68004E53HA |
|----------------|------------|

| | |
|--------------|-------------------------|
| Disk Status: | Online, 17928192 blocks |
|--------------|-------------------------|

| | | | |
|-----|-------------|----------------|----------------|
| 0:2 | Vendor: IBM | Model: DRVS09D | Revision: 0270 |
|-----|-------------|----------------|----------------|

| | |
|----------------|------------|
| Serial Number: | 13013935HA |
|----------------|------------|

| | |
|--------------|-------------------------|
| Disk Status: | Online, 17928192 blocks |
|--------------|-------------------------|

| | | | |
|-----|-------------|----------------|----------------|
| 0:3 | Vendor: IBM | Model: DRVS09D | Revision: 0270 |
|-----|-------------|----------------|----------------|

| | |
|----------------|------------|
| Serial Number: | 13016897HA |
|----------------|------------|

| | |
|--------------|-------------------------|
| Disk Status: | Online, 17928192 blocks |
|--------------|-------------------------|

| | | | |
|-----|-------------|----------------|----------------|
| 0:4 | Vendor: IBM | Model: DRVS09D | Revision: 0270 |
|-----|-------------|----------------|----------------|

| | |
|----------------|------------|
| Serial Number: | 68019905HA |
|----------------|------------|

| | |
|--------------|-------------------------|
| Disk Status: | Online, 17928192 blocks |
|--------------|-------------------------|

| | | | |
|-----|-------------|----------------|----------------|
| 0:5 | Vendor: IBM | Model: DRVS09D | Revision: 0270 |
|-----|-------------|----------------|----------------|

| | |
|----------------|------------|
| Serial Number: | 68012753HA |
|----------------|------------|

| | |
|--------------|-------------------------|
| Disk Status: | Online, 17928192 blocks |
|--------------|-------------------------|

| | | | |
|-----|-----------------|--------------------|----------------|
| 0:6 | Vendor: ESG-SHV | Model: SCA HSBP M6 | Revision: 0.61 |
|-----|-----------------|--------------------|----------------|

Logical Drives:

/dev/rd/c0d0: RAID-5, Online, 89640960 blocks, Write Thru

No Rebuild or Consistency Check in Progress

To simplify the monitoring process for custom software, the special file

/proc/rd/status returns "OK" when all DAC960 controllers in the system are operating normally and no failures have occurred, or "ALERT" if any logical drives are offline or critical or any non-standby physical drives are dead.

Configuration commands for controller N are available via the special file /proc/rd/cN/user_command. A human readable command can be written to this special file to initiate a configuration operation, and the results of the operation can then be read back from the special file in addition to being logged to the system console. The shell command sequence

```
echo "<configuration-command>" > /proc/rd/c0/user_command
cat /proc/rd/c0/user_command
```

is typically used to execute configuration commands. The configuration commands are:

flush-cache

The "flush-cache" command flushes the controller's cache. The system automatically flushes the cache at shutdown or if the driver module is unloaded, so this command is only needed to be certain a write back cache is flushed to disk before the system is powered off by a command to a UPS. Note that the flush-cache command also stops an asynchronous rebuild or consistency check, so it should not be used except when the system is being halted.

kill <channel>:<target-id>

The "kill" command marks the physical drive <channel>:<target-id> as DEAD. This command is provided primarily for testing, and should not be used during normal system operation.

make-online <channel>:<target-id>

The "make-online" command changes the physical drive <channel>:<target-id> from status DEAD to status ONLINE. In cases where multiple physical drives have been killed simultaneously, this command may be used to bring all but one of them back online, after which a rebuild to the final drive is necessary.

Warning: make-online should only be used on a dead physical drive that is an active part of a drive group, never on a standby drive. The command should never be used on a dead drive that is part of a critical logical drive; rebuild should be used if only a single drive is dead.

make-standby <channel>:<target-id>

The "make-standby" command changes physical drive <channel>:<target-id> from status DEAD to status STANDBY. It should only be used in cases where a dead drive was replaced after an automatic rebuild was performed onto a standby drive. It cannot be used to add a standby drive to the controller configuration if one was not created initially; the BIOS Configuration Utility must be used for that currently.

rebuild <channel>:<target-id>

README.DAC960.txt

The "rebuild" command initiates an asynchronous rebuild onto physical drive <channel>:<target-id>. It should only be used when a dead drive has been replaced.

check-consistency <logical-drive-number>

The "check-consistency" command initiates an asynchronous consistency check of <logical-drive-number> with automatic restoration. It can be used whenever it is desired to verify the consistency of the redundancy information.

cancel-rebuild

cancel-consistency-check

The "cancel-rebuild" and "cancel-consistency-check" commands cancel any rebuild or consistency check operations previously initiated.

EXAMPLE I - DRIVE FAILURE WITHOUT A STANDBY DRIVE

The following annotated logs demonstrate the controller configuration and online status monitoring capabilities of the Linux DAC960 Driver. The test configuration comprises 6 1GB Quantum Atlas I disk drives on two channels of a DAC960PJ controller. The physical drives are configured into a single drive group without a standby drive, and the drive group has been configured into two logical drives, one RAID-5 and one RAID-6. Note that these logs are from an earlier version of the driver and the messages have changed somewhat with newer releases, but the functionality remains similar. First, here is the current status of the RAID configuration:

```
gwynedd:/u/lnz# cat /proc/rd/c0/current_status
***** DAC960 RAID Driver Version 2.0.0 of 23 March 1999 *****
Copyright 1998-1999 by Leonard N. Zubkoff <lnz@dandelion.com>
Configuring Mylex DAC960PJ PCI RAID Controller
Firmware Version: 4.06-0-08, Channels: 3, Memory Size: 8MB
PCI Bus: 0, Device: 19, Function: 1, I/O Address: Unassigned
PCI Address: 0xFD4FC000 mapped at 0x8807000, IRQ Channel: 9
Controller Queue Depth: 128, Maximum Blocks per Command: 128
Driver Queue Depth: 127, Maximum Scatter/Gather Segments: 33
Stripe Size: 64KB, Segment Size: 8KB, BIOS Geometry: 255/63
Physical Devices:
  0:1 - Disk: Online, 2201600 blocks
  0:2 - Disk: Online, 2201600 blocks
  0:3 - Disk: Online, 2201600 blocks
  1:1 - Disk: Online, 2201600 blocks
  1:2 - Disk: Online, 2201600 blocks
  1:3 - Disk: Online, 2201600 blocks
Logical Drives:
  /dev/rd/c0d0: RAID-5, Online, 5498880 blocks, Write Thru
  /dev/rd/c0d1: RAID-6, Online, 3305472 blocks, Write Thru
No Rebuild or Consistency Check in Progress
```

```
gwynedd:/u/lnz# cat /proc/rd/status
OK
```

The above messages indicate that everything is healthy, and /proc/rd/status

README.DAC960.txt

returns "OK" indicating that there are no problems with any DAC960 controller in the system. For demonstration purposes, while I/O is active Physical Drive 1:1 is now disconnected, simulating a drive failure. The failure is noted by the driver within 10 seconds of the controller's having detected it, and the driver logs the following console status messages indicating that Logical Drives 0 and 1 are now CRITICAL as a result of Physical Drive 1:1 being DEAD:

```
DAC960#0: Physical Drive 1:2 Error Log: Sense Key = 6, ASC = 29, ASCQ = 02
DAC960#0: Physical Drive 1:3 Error Log: Sense Key = 6, ASC = 29, ASCQ = 02
DAC960#0: Physical Drive 1:1 killed because of timeout on SCSI command
DAC960#0: Physical Drive 1:1 is now DEAD
DAC960#0: Logical Drive 0 (/dev/rd/c0d0) is now CRITICAL
DAC960#0: Logical Drive 1 (/dev/rd/c0d1) is now CRITICAL
```

The Sense Keys logged here are just Check Condition / Unit Attention conditions arising from a SCSI bus reset that is forced by the controller during its error recovery procedures. Concurrently with the above, the driver status available from /proc/rd also reflects the drive failure. The status message in /proc/rd/status has changed from "OK" to "ALERT":

```
gwynedd:/u/lnz# cat /proc/rd/status
ALERT
```

and /proc/rd/c0/current_status has been updated:

```
gwynedd:/u/lnz# cat /proc/rd/c0/current_status
```

```
...
```

Physical Devices:

```
0:1 - Disk: Online, 2201600 blocks
0:2 - Disk: Online, 2201600 blocks
0:3 - Disk: Online, 2201600 blocks
1:1 - Disk: Dead, 2201600 blocks
1:2 - Disk: Online, 2201600 blocks
1:3 - Disk: Online, 2201600 blocks
```

Logical Drives:

```
/dev/rd/c0d0: RAID-5, Critical, 5498880 blocks, Write Thru
/dev/rd/c0d1: RAID-6, Critical, 3305472 blocks, Write Thru
```

No Rebuild or Consistency Check in Progress

Since there are no standby drives configured, the system can continue to access the logical drives in a performance degraded mode until the failed drive is replaced and a rebuild operation completed to restore the redundancy of the logical drives. Once Physical Drive 1:1 is replaced with a properly functioning drive, or if the physical drive was killed without having failed (e.g., due to electrical problems on the SCSI bus), the user can instruct the controller to initiate a rebuild operation onto the newly replaced drive:

```
gwynedd:/u/lnz# echo "rebuild 1:1" > /proc/rd/c0/user_command
gwynedd:/u/lnz# cat /proc/rd/c0/user_command
Rebuild of Physical Drive 1:1 Initiated
```

The echo command instructs the controller to initiate an asynchronous rebuild operation onto Physical Drive 1:1, and the status message that results from the operation is then available for reading from /proc/rd/c0/user_command, as well as being logged to the console by the driver.

README.DAC960.txt

Within 10 seconds of this command the driver logs the initiation of the asynchronous rebuild operation:

```
DAC960#0: Rebuild of Physical Drive 1:1 Initiated
DAC960#0: Physical Drive 1:1 Error Log: Sense Key = 6, ASC = 29, ASCQ = 01
DAC960#0: Physical Drive 1:1 is now WRITE-ONLY
DAC960#0: Rebuild in Progress: Logical Drive 0 (/dev/rd/c0d0) 1% completed
```

and /proc/rd/c0/current_status is updated:

```
gwynedd:/u/lnz# cat /proc/rd/c0/current_status
```

```
...
Physical Devices:
 0:1 - Disk: Online, 2201600 blocks
 0:2 - Disk: Online, 2201600 blocks
 0:3 - Disk: Online, 2201600 blocks
 1:1 - Disk: Write-Only, 2201600 blocks
 1:2 - Disk: Online, 2201600 blocks
 1:3 - Disk: Online, 2201600 blocks
Logical Drives:
 /dev/rd/c0d0: RAID-5, Critical, 5498880 blocks, Write Thru
 /dev/rd/c0d1: RAID-6, Critical, 3305472 blocks, Write Thru
Rebuild in Progress: Logical Drive 0 (/dev/rd/c0d0) 6% completed
```

As the rebuild progresses, the current status in /proc/rd/c0/current_status is updated every 10 seconds:

```
gwynedd:/u/lnz# cat /proc/rd/c0/current_status
```

```
...
Physical Devices:
 0:1 - Disk: Online, 2201600 blocks
 0:2 - Disk: Online, 2201600 blocks
 0:3 - Disk: Online, 2201600 blocks
 1:1 - Disk: Write-Only, 2201600 blocks
 1:2 - Disk: Online, 2201600 blocks
 1:3 - Disk: Online, 2201600 blocks
Logical Drives:
 /dev/rd/c0d0: RAID-5, Critical, 5498880 blocks, Write Thru
 /dev/rd/c0d1: RAID-6, Critical, 3305472 blocks, Write Thru
Rebuild in Progress: Logical Drive 0 (/dev/rd/c0d0) 15% completed
```

and every minute a progress message is logged to the console by the driver:

```
DAC960#0: Rebuild in Progress: Logical Drive 0 (/dev/rd/c0d0) 32% completed
DAC960#0: Rebuild in Progress: Logical Drive 0 (/dev/rd/c0d0) 63% completed
DAC960#0: Rebuild in Progress: Logical Drive 0 (/dev/rd/c0d0) 94% completed
DAC960#0: Rebuild in Progress: Logical Drive 1 (/dev/rd/c0d1) 94% completed
```

Finally, the rebuild completes successfully. The driver logs the status of the logical and physical drives and the rebuild completion:

```
DAC960#0: Rebuild Completed Successfully
DAC960#0: Physical Drive 1:1 is now ONLINE
DAC960#0: Logical Drive 0 (/dev/rd/c0d0) is now ONLINE
DAC960#0: Logical Drive 1 (/dev/rd/c0d1) is now ONLINE
```

README.DAC960.txt

/proc/rd/c0/current_status is updated:

```
gwynedd:/u/lnz# cat /proc/rd/c0/current_status
```

```
...
```

```
Physical Devices:
```

```
0:1 - Disk: Online, 2201600 blocks
0:2 - Disk: Online, 2201600 blocks
0:3 - Disk: Online, 2201600 blocks
1:1 - Disk: Online, 2201600 blocks
1:2 - Disk: Online, 2201600 blocks
1:3 - Disk: Online, 2201600 blocks
```

```
Logical Drives:
```

```
/dev/rd/c0d0: RAID-5, Online, 5498880 blocks, Write Thru
/dev/rd/c0d1: RAID-6, Online, 3305472 blocks, Write Thru
```

```
Rebuild Completed Successfully
```

and /proc/rd/status indicates that everything is healthy once again:

```
gwynedd:/u/lnz# cat /proc/rd/status
```

```
OK
```

EXAMPLE II - DRIVE FAILURE WITH A STANDBY DRIVE

The following annotated logs demonstrate the controller configuration and online status monitoring capabilities of the Linux DAC960 Driver. The test configuration comprises 6 1GB Quantum Atlas I disk drives on two channels of a DAC960PJ controller. The physical drives are configured into a single drive group with a standby drive, and the drive group has been configured into two logical drives, one RAID-5 and one RAID-6. Note that these logs are from an earlier version of the driver and the messages have changed somewhat with newer releases, but the functionality remains similar. First, here is the current status of the RAID configuration:

```
gwynedd:/u/lnz# cat /proc/rd/c0/current_status
```

```
***** DAC960 RAID Driver Version 2.0.0 of 23 March 1999 *****
```

```
Copyright 1998-1999 by Leonard N. Zubkoff <lnz@dandelion.com>
```

```
Configuring Mylex DAC960PJ PCI RAID Controller
```

```
Firmware Version: 4.06-0-08, Channels: 3, Memory Size: 8MB
```

```
PCI Bus: 0, Device: 19, Function: 1, I/O Address: Unassigned
```

```
PCI Address: 0xFD4FC000 mapped at 0x8807000, IRQ Channel: 9
```

```
Controller Queue Depth: 128, Maximum Blocks per Command: 128
```

```
Driver Queue Depth: 127, Maximum Scatter/Gather Segments: 33
```

```
Stripe Size: 64KB, Segment Size: 8KB, BIOS Geometry: 255/63
```

```
Physical Devices:
```

```
0:1 - Disk: Online, 2201600 blocks
0:2 - Disk: Online, 2201600 blocks
0:3 - Disk: Online, 2201600 blocks
1:1 - Disk: Online, 2201600 blocks
1:2 - Disk: Online, 2201600 blocks
1:3 - Disk: Standby, 2201600 blocks
```

```
Logical Drives:
```

```
/dev/rd/c0d0: RAID-5, Online, 4399104 blocks, Write Thru
```

```
/dev/rd/c0d1: RAID-6, Online, 2754560 blocks, Write Thru
```

```
No Rebuild or Consistency Check in Progress
```

README.DAC960.txt

```
gwynedd:/u/lnz# cat /proc/rd/status
OK
```

The above messages indicate that everything is healthy, and /proc/rd/status returns "OK" indicating that there are no problems with any DAC960 controller in the system. For demonstration purposes, while I/O is active Physical Drive 1:2 is now disconnected, simulating a drive failure. The failure is noted by the driver within 10 seconds of the controller's having detected it, and the driver logs the following console status messages:

```
DAC960#0: Physical Drive 1:1 Error Log: Sense Key = 6, ASC = 29, ASCQ = 02
DAC960#0: Physical Drive 1:3 Error Log: Sense Key = 6, ASC = 29, ASCQ = 02
DAC960#0: Physical Drive 1:2 killed because of timeout on SCSI command
DAC960#0: Physical Drive 1:2 is now DEAD
DAC960#0: Physical Drive 1:2 killed because it was removed
DAC960#0: Logical Drive 0 (/dev/rd/c0d0) is now CRITICAL
DAC960#0: Logical Drive 1 (/dev/rd/c0d1) is now CRITICAL
```

Since a standby drive is configured, the controller automatically begins rebuilding onto the standby drive:

```
DAC960#0: Physical Drive 1:3 is now WRITE-ONLY
DAC960#0: Rebuild in Progress: Logical Drive 0 (/dev/rd/c0d0) 4% completed
```

Concurrently with the above, the driver status available from /proc/rd also reflects the drive failure and automatic rebuild. The status message in /proc/rd/status has changed from "OK" to "ALERT":

```
gwynedd:/u/lnz# cat /proc/rd/status
ALERT
```

and /proc/rd/c0/current_status has been updated:

```
gwynedd:/u/lnz# cat /proc/rd/c0/current_status
```

```
...
Physical Devices:
 0:1 - Disk: Online, 2201600 blocks
 0:2 - Disk: Online, 2201600 blocks
 0:3 - Disk: Online, 2201600 blocks
 1:1 - Disk: Online, 2201600 blocks
 1:2 - Disk: Dead, 2201600 blocks
 1:3 - Disk: Write-Only, 2201600 blocks
Logical Drives:
 /dev/rd/c0d0: RAID-5, Critical, 4399104 blocks, Write Thru
 /dev/rd/c0d1: RAID-6, Critical, 2754560 blocks, Write Thru
Rebuild in Progress: Logical Drive 0 (/dev/rd/c0d0) 4% completed
```

As the rebuild progresses, the current status in /proc/rd/c0/current_status is updated every 10 seconds:

```
gwynedd:/u/lnz# cat /proc/rd/c0/current_status
```

```
...
Physical Devices:
 0:1 - Disk: Online, 2201600 blocks
 0:2 - Disk: Online, 2201600 blocks
 0:3 - Disk: Online, 2201600 blocks
```

README.DAC960.txt

```
1:1 - Disk: Online, 2201600 blocks
1:2 - Disk: Dead, 2201600 blocks
1:3 - Disk: Write-Only, 2201600 blocks
Logical Drives:
/dev/rd/c0d0: RAID-5, Critical, 4399104 blocks, Write Thru
/dev/rd/c0d1: RAID-6, Critical, 2754560 blocks, Write Thru
Rebuild in Progress: Logical Drive 0 (/dev/rd/c0d0) 40% completed
```

and every minute a progress message is logged on the console by the driver:

```
DAC960#0: Rebuild in Progress: Logical Drive 0 (/dev/rd/c0d0) 40% completed
DAC960#0: Rebuild in Progress: Logical Drive 0 (/dev/rd/c0d0) 76% completed
DAC960#0: Rebuild in Progress: Logical Drive 1 (/dev/rd/c0d1) 66% completed
DAC960#0: Rebuild in Progress: Logical Drive 1 (/dev/rd/c0d1) 84% completed
```

Finally, the rebuild completes successfully. The driver logs the status of the logical and physical drives and the rebuild completion:

```
DAC960#0: Rebuild Completed Successfully
DAC960#0: Physical Drive 1:3 is now ONLINE
DAC960#0: Logical Drive 0 (/dev/rd/c0d0) is now ONLINE
DAC960#0: Logical Drive 1 (/dev/rd/c0d1) is now ONLINE
```

/proc/rd/c0/current_status is updated:

```
***** DAC960 RAID Driver Version 2.0.0 of 23 March 1999 *****
Copyright 1998-1999 by Leonard N. Zubkoff <lnz@dandelion.com>
Configuring Mylex DAC960PJ PCI RAID Controller
Firmware Version: 4.06-0-08, Channels: 3, Memory Size: 8MB
PCI Bus: 0, Device: 19, Function: 1, I/O Address: Unassigned
PCI Address: 0xFD4FC000 mapped at 0x8807000, IRQ Channel: 9
Controller Queue Depth: 128, Maximum Blocks per Command: 128
Driver Queue Depth: 127, Maximum Scatter/Gather Segments: 33
Stripe Size: 64KB, Segment Size: 8KB, BIOS Geometry: 255/63
Physical Devices:
0:1 - Disk: Online, 2201600 blocks
0:2 - Disk: Online, 2201600 blocks
0:3 - Disk: Online, 2201600 blocks
1:1 - Disk: Online, 2201600 blocks
1:2 - Disk: Dead, 2201600 blocks
1:3 - Disk: Online, 2201600 blocks
Logical Drives:
/dev/rd/c0d0: RAID-5, Online, 4399104 blocks, Write Thru
/dev/rd/c0d1: RAID-6, Online, 2754560 blocks, Write Thru
Rebuild Completed Successfully
```

and /proc/rd/status indicates that everything is healthy once again:

```
gwynedd:/u/lnz# cat /proc/rd/status
OK
```

Note that the absence of a viable standby drive does not create an "ALERT" status. Once dead Physical Drive 1:2 has been replaced, the controller must be told that this has occurred and that the newly replaced drive should become the new standby drive:

README.DAC960.txt

```
gwynedd:/u/lnz# echo "make-standby 1:2" > /proc/rd/c0/user_command
gwynedd:/u/lnz# cat /proc/rd/c0/user_command
Make Standby of Physical Drive 1:2 Succeeded
```

The echo command instructs the controller to make Physical Drive 1:2 into a standby drive, and the status message that results from the operation is then available for reading from /proc/rd/c0/user_command, as well as being logged to the console by the driver. Within 60 seconds of this command the driver logs:

```
DAC960#0: Physical Drive 1:2 Error Log: Sense Key = 6, ASC = 29, ASCQ = 01
DAC960#0: Physical Drive 1:2 is now STANDBY
DAC960#0: Make Standby of Physical Drive 1:2 Succeeded
```

and /proc/rd/c0/current_status is updated:

```
gwynedd:/u/lnz# cat /proc/rd/c0/current_status
...
Physical Devices:
 0:1 - Disk: Online, 2201600 blocks
 0:2 - Disk: Online, 2201600 blocks
 0:3 - Disk: Online, 2201600 blocks
 1:1 - Disk: Online, 2201600 blocks
 1:2 - Disk: Standby, 2201600 blocks
 1:3 - Disk: Online, 2201600 blocks
Logical Drives:
 /dev/rd/c0d0: RAID-5, Online, 4399104 blocks, Write Thru
 /dev/rd/c0d1: RAID-6, Online, 2754560 blocks, Write Thru
Rebuild Completed Successfully
```