

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Visit our website - *Click HERE*



A Division of ELMA Electronic Inc.

RAIDStor

Installation and Use

ACT/Technico Model Number 6210 & 6211
ACT/Technico P/N 02-D303-00 Rev - 1

Revision 1.4
12/23/09

Revision Table		
Revision	**Description**	**Date**
0.1	Initial Draft	03/2006
0.2	Single Blade Updates	07/14/06
0.3	Merge Single/Dual Blade Web Pages	08/17/06
0.4	Remove Network RAID Page	09/06/06
1.0	First Release	05/11/07
1.1	Add Software Installation Procedure	5/30/07
1.2	Add Performance Section	9/24/07
1.3	Update Console Cable Pin-out	12/19/07
1.4	Update for 6211 RAIDStor SATA Shuttle Release Console Cable update, new cable uses different color pairs. Add Software Revision History	12/23/09

Notice

This manual is provided for information purposes only and is subject to change without notice. ACT/Technico has taken reasonable efforts to provide an accurate document, and assumes no liability for any inaccuracies.

ACT/Technico, a division of ELMA Electronic, Inc.
760 Veterans Circle
Warminster, Pa. 18974

Table of Contents - Brief

1 OVERVIEW 10

1.1 PREFACE 10
1.2 INTRODUCTION 10
1.3 PHOTOS 11
1.4 BLOCK DIAGRAM AND BOARD LAYOUT 15
1.5 STANDARDS REFERENCES 17

2 GENERAL DESCRIPTION 18

2.1 RAIDSTOR HARDWARE DESCRIPTION 18
2.2 RAIDSTOR SOFTWARE DESCRIPTION 20
2.3 SINGLE BLADE RAIDSTOR 21
2.4 DUAL BLADE RAIDSTOR 21
2.5 FRONT PANEL LEDS 22
2.6 OPERATING SYSTEM 24
2.7 COMPACT FLASH 24
2.8 REQUIRED LINUX MODULES 24
2.9 WEB INTERFACE 24
2.10 SNMP INTERFACE 24

3 RAIDSTOR USAGE AND OPERATION 25

3.1 INITIAL INSTALLATION 25
3.2 SHUTTLE REMOVAL/INSERTION PROCESS 25
3.3 RAIDSTOR SHUTDOWN 26
3.4 DEFAULT PASSWORDS 27
3.5 DEFAULT IP ADDRESS ASSIGNMENTS 27
3.6 RAID PARTITIONING 28
3.7 NETWORKING 30
3.8 HIGH AVAILABILITY SERVICES 36
3.9 USB 41
3.10 FIRMWARE 43
3.11 TESTING THE RAIDSTOR 45

4 CONTROL AND STATUS INTERFACES 47

4.1 COMMAND LINE INTERFACE 47
4.2 WEB INTERFACE 47
4.3 SNMP 63
4.4 SECURITY 64

5 DUAL BLADE CONFIGURATION PROCEDURES USING THE COMMAND LINE 66

5.1 STEP BY STEP SHUTDOWN OF HEARTBEAT AND DRBD 66
5.2 STEP BY STEP STARTUP OF HEARTBEAT AND DRBD 66
5.3 CHANGE THE IP PARAMETERS OF A RAIDSTOR BLADE FROM /ETC/HOSTS 67
5.4 ADD A SERVICE TO HEARTBEAT 67
5.5 ADD A DRBD DISK 68

6 RAIDSTOR SOFTWARE INSTALLATION 73

6.1 DHCP SERVER SETUP 73
6.2 INSTALLING THE SOFTWARE 75
6.3 POST INSTALLATION PROCEDURE 76

7 RAIDSTOR SUSTAINED NFS PERFORMANCE 77

8 ENVIRONMENTAL, SAFETY AND RELIABILITY SPECIFICATIONS 78

8.1 POWER 78
8.2 SAFETY 78
8.3 ENVIRONMENTAL SPECIFICATIONS RAIDSTOR BASEBOARD ONLY – NO DRIVES 78

8.4 RELIABILITY 78

9 MECHANICAL SPECIFICATIONS – CONNECTOR PIN ASSIGNMENTS 80

9.1 BOARD MECHANICS 80
9.2 SHUTTLE 6210SH & 6211SH 91
9.3 CONSOLE CABLE 94

10 MEMORY MAP 95

10.1 PCI/ISA I/O SPACE MEMORY MAP 95

11 ACT/TECHNICO MIB(S) 97

11.1 MIB CONVENTIONS 97
11.2 ACT/TECHNICO CORPORATE MIB 97
11.3 ACT/TECHNICO RAIDSTOR MIB 98

12 RAIDSTOR SOFTWARE REVISION HISTORY 121

12.1 SOFTWARE REVISION DETAILS 121

Table of Contents - Complete

1 OVERVIEW 10

1.1 PREFACE 10
1.2 INTRODUCTION 10
1.3 PHOTOS 11
1.4 BLOCK DIAGRAM AND BOARD LAYOUT 15
1.5 STANDARDS REFERENCES 17

2 GENERAL DESCRIPTION 18

2.1 RAIDSTOR HARDWARE DESCRIPTION 18
2.1.1 Key RAIDStor Hardware Features 18
2.2 RAIDSTOR SOFTWARE DESCRIPTION 20
2.2.1 Key RAIDStor Software Features 20
2.3 SINGLE BLADE RAIDSTOR 21
2.4 DUAL BLADE RAIDSTOR 21
2.5 FRONT PANEL LEDS 22
2.5.1 Drive LEDS 23
2.6 OPERATING SYSTEM 24
2.7 COMPACT FLASH 24
2.8 REQUIRED LINUX MODULES 24
2.9 WEB INTERFACE 24
2.10 SNMP INTERFACE 24

3 RAIDSTOR USAGE AND OPERATION 25

3.1 INITIAL INSTALLATION 25
3.2 SHUTTLE REMOVAL/INSERTION PROCESS 25
3.2.1 Dual Blade RAIDStor Failover Policy 26
3.3 RAIDSTOR SHUTDOWN 26
3.4 DEFAULT PASSWORDS 27
3.5 DEFAULT IP ADDRESS ASSIGNMENTS 27
3.5.1 Single Blade RAIDStor 27
3.5.2 Dual Blade RAIDStor 27
3.5.3 Changing the IP Address 28
3.5.3.1 Via Webpage 28
3.5.3.2 Via SNMP 28
3.5.3.3 Via /etc/hosts 28
3.6 RAID PARTITIONING 28
3.6.1 Root File System RAID Partitions 28
3.6.2 Single Blade RAIDStor Partitions 28
3.6.3 Dual Blade RAIDStor Partitions 29
3.7 NETWORKING 30
3.7.1 Bonding Driver 30
3.7.1.1 Network Interface Options 30
3.7.1.2 Bonding Script 31
3.7.1.2.1 Loading the Bonding Driver Module 31
3.7.1.2.2 Initializing the Bonding Driver Network Interface 33
3.7.1.2.3 Enslaving the Ethernet Devices to the Bonding Driver 33
3.7.1.2.4 Setting the Default Route 33
3.7.2 File System Exports with NFS and SMB/CIFS 33
3.7.2.1 NFS Export Options 33
3.7.2.2 NFS User ID Mapping 35
3.7.2.3 Default NFS Options 35
3.7.2.4 SMB/CIFS Export Options 35
3.7.2.5 Typical RAIDStor smb.conf File 35
3.7.2.6 Restarting Samba 36
3.8 HIGH AVAILABILITY SERVICES 36
3.8.1 Distributed Replicated Block Device (DRBD) 36

3.8.1.1 Quick Summary of How DRBD Works ... 37
3.8.1.2 DRBD Configuration ... 37
3.8.1.3 Controlling DRBD ... 37
3.8.1.3.1 Stopping, Starting, Loading, and Unloading DRBD ... 37
3.8.1.3.2 Forcing DRBD resynchronization ... 38
3.8.1.3.3 Booting a RAIDStor with DRBD Disabled ... 38
3.8.1.3.4 Layout of the DRBD Disk ... 38
3.8.1.3.5 Mounting DRBD Disks in the Local File System ... 38
3.8.2 Heartbeat ... 39
3.8.2.1 Heartbeat Configuration ... 40
3.8.2.2 Heartbeat Resources ... 40
3.8.2.2.1 IPaddr Script ... 40
3.8.2.2.2 DRBD Script ... 40
3.8.2.2.3 Filesystem Script ... 40
3.8.2.2.4 NFS and SMB/CIFS Server Scripts ... 41
3.8.2.2.5 led_nas Script ... 41
3.8.2.3 Heartbeat Authentication Keys ... 41
3.9 USB ... 41
3.10 FIRMWARE ... 43
3.10.1 Key Boot Options ... 44
3.11 TESTING THE RAIDSTOR ... 45
3.11.1 Single Blade RAIDStor ... 45
3.11.1.1 Initial Setup ... 45
3.11.1.2 Shuttle Hotswap Test ... 45
3.11.2 Dual Blade RAIDStor ... 45
3.11.2.1 Initial Setup ... 45
3.11.2.2 Shuttle Hotswap Test ... 45
3.11.2.3 Blade Failover Test ... 46

4 CONTROL AND STATUS INTERFACES ... 47

4.1 COMMAND LINE INTERFACE ... 47
4.1.1 Serial Console Port ... 47
4.1.2 Telnet/Rlogin/SSH ... 47
4.2 WEB INTERFACE ... 47
4.2.1 Home Page (Login) ... 48
4.2.2 System Page ... 49
4.2.2.1 System ... 50
4.2.2.2 System Health Summary ... 51
4.2.2.2.1 Single Blade ... 51
4.2.2.2.2 Dual Blade ... 51
4.2.3 Status Page ... 52
4.2.4 Network Settings Page ... 54
4.2.5 Blade Info Page ... 56
4.2.6 Admin Account Page ... 58
4.2.7 System Time Page ... 59
4.2.8 Email Notification Page ... 60
4.2.9 Trap Notification Page ... 61
4.2.10 SNMP Web Page ... 62
4.3 SNMP ... 63
4.3.1 SNMP Agent ... 63
4.3.1.1 Normal Operation ... 63
4.3.1.2 SNMP Set Operation ... 63
4.3.2 Traps and Email Notification ... 63
4.4 SECURITY ... 64
4.4.1 Login Security ... 64
4.4.2 DRBD Security ... 64
4.4.3 Server Security ... 64
4.4.4 Web Page Security ... 65

4.4.5 SNMP Security *65*

5 DUAL BLADE CONFIGURATION PROCEDURES USING THE COMMAND LINE 66

5.1 STEP BY STEP SHUTDOWN OF HEARTBEAT AND DRBD 66
5.2 STEP BY STEP STARTUP OF HEARTBEAT AND DRBD 66
5.3 CHANGE THE IP PARAMETERS OF A RAIDSTOR BLADE FROM /ETC/HOSTS 67
5.4 ADD A SERVICE TO HEARTBEAT 67
5.5 ADD A DRBD DISK 68
5.5.1 Shutdown Heartbeat and DRBD *68*
5.5.2 Change the Run Level *68*
5.5.3 Repartition the Disks *68*
5.5.4 Reboot the Blades *69*
5.5.5 Make a New RAID *69*
5.5.6 Modify the DRBD Configuration *69*
5.5.7 Modify mountpoint Configuration *70*
5.5.8 Modify NFS or SMB Configuration *70*
5.5.9 Modify the Heartbeat Configuration *71*
5.5.10 Change the Run Level *71*
5.5.11 Start Bonding Driver and DRBD *71*
5.5.12 Create a Filesystem on the DRBD disk *71*
5.5.13 Start Heartbeat *71*
5.5.14 Set Access Permissions *72*

6 RAIDSTOR SOFTWARE INSTALLATION 73

6.1 DHCP SERVER SETUP 73
6.1.1 Record the MAC address of Target RAIDStor *73*
6.1.2 DHCP Configuration *74*
6.1.2.1 Single Blade or Primary Dual Blade DHCP Setup 74
6.1.2.2 Backup Blade DHCP Setup 75
6.1.3 MenMon Configuration *75*
6.2 INSTALLING THE SOFTWARE 75
6.3 POST INSTALLATION PROCEDURE 76

7 RAIDSTOR SUSTAINED NFS PERFORMANCE 77

8 ENVIRONMENTAL, SAFETY AND RELIABILITY SPECIFICATIONS 78

8.1 POWER 78
8.2 SAFETY 78
8.3 ENVIRONMENTAL SPECIFICATIONS RAIDSTOR BASEBOARD ONLY – NO DRIVES 78
8.4 RELIABILITY 78
8.4.1 MTBF – MIL-STD-217 Ground Benign @ 40°C *79*

9 MECHANICAL SPECIFICATIONS – CONNECTOR PIN ASSIGNMENTS 80

9.1 BOARD MECHANICS 80
9.1.1 CD-ROM Connector P25 ..... *80*
9.1.2 CD-ROM Adapter *81*
9.1.2.1 Connector P24 81
9.1.2.2 Connector J4 82
9.1.3 CompactFlash Connector P28 ..... *83*
9.1.4 Disk Shuttle Carrier Connector J6 ..... *84*
9.1.5 Connector for connecting shuttle P26 and P27 ..... *85*
9.1.6 USB Connector J7 ..... *86*
9.1.7 COM1 Connector P7 ..... *86*
9.1.8 Connector J1 ..... *87*
9.1.9 Connector J2 ..... *88*
9.1.10 Connector J3 ..... *89*
9.1.11 Connector J5 ..... *90*
9.2 SHUTTLE 6210SH & 6211SH 91

9.2.1 6210SH PATA Connector J1 ..... *91*
9.2.2 6211SH SATA Connector J1 ..... *92*
9.2.3 6210SH and 6211SH Connector J2 ..... *93*
9.3 CONSOLE CABLE 94
10 MEMORY MAP 95
10.1 PCI/ISA I/O SPACE MEMORY MAP 95
11 ACT/TECHNICO MIB(S) 97
11.1 MIB CONVENTIONS 97
11.2 ACT/TECHNICO CORPORATE MIB 97
11.3 ACT/TECHNICO RAIDSTOR MIB 98
12 RAIDSTOR SOFTWARE REVISION HISTORY 121
12.1 SOFTWARE REVISION DETAILS 121

List of Figures

Figure 2: 4 Shuttle RAIDStor *11*
Figure 1: 2 Shuttle 6210 RAIDStor *11*
Figure 4: Rear Transition Module *12*
Figure 3: 4 Shuttle RAIDStor with DVD *12*
Figure 5: RAIDStor w/ 6210SH shuttles removed *13*
Figure 6: 6210SH Drive Shuttle *13*
Figure 7: RAIDStor with 6211 Shuttles Removed *14*
Figure 8: 6211 SATA Shuttle *14*
Figure 9: RAIDStor Block Diagram *15*
Figure 10: 6210 RAIDStor Hardware Component Layout *16*
Figure 11: Front Panel LEDs *23*
Figure 12: Shuttle Insertion/Removal *25*
Figure 13: Web Page – Home *48*
Figure 14: Web Page – System – Single Blade *49*
Figure 15: Web Page – System -- Dual Blade *50*
Figure 16: Web Page – Status – Single Blade *52*
Figure 17: Web Page -- Status -- Dual Blade *53*
Figure 18: Web Page - Network Settings – Single Blade *54*
Figure 19: Web Page -- Network Settings -- Dual Blade *55*
Figure 20: Web Page – Blade Info – Single Blade *56*
Figure 21: Web Page -- Blade Info -- Dual Blade *57*
Figure 22: Web Page – Administrative Account *58*
Figure 23: Web Page - System Time *59*
Figure 24: Web Page - Email Notification *60*
Figure 25: Web Page - Trap Notification *61*
Figure 26: Web Page - SNMP *62*

List of Tables

Table 01: Standards References *17*
Table 02: Key RAIDStor Software Features *20*
Table 03: Single Blade RAID Levels *21*
Table 04: Dual Blade RAID Levels *22*
Table 05: Typical RAID Partitions 2 Drives Single Blade *29*
Table 06: Typical RAID Partitions 4 Drives Single Blade *29*
Table 07: Typical RAID Partitions 2 Drives Dual Blade *30*
Table 08: Typical RAID Partitions 4 Drives Dual Blade *30*
Table 09: Bonding Driver Policy *31*
Table 10: Bonding Driver Options *32*
Table 11: NFS Clients *34*
Table 12: NFS Options *34*
Table 13: NFS User ID Mapping *35*
Table 14: DRDB Mount Options *39*
Table 15: SNMP Traps *64*
Table 16: RAIDStor Installation Command Table *75*
Table 17: Sustained NFS Performance *77*
Table 18: Pin Assignment of the 44-Pin CD-ROM Plug Connector P25 ..... *80*
Table 19: Signal Mnemonics of 44-pin Plug Connector for CD-ROM *80*
Table 20: Pin Assignment of the 44-Pin CD-ROM Plug Connector P24 ..... *81*
Table 21: Pin Assignment of the 50-Pin CD-ROM Connector *82*
Table 22: Pin Assignment of the 50-pin CompactFlash Connector *83*
Table 23: Pin Assignment of the 40-pin Connector *84*
Table 24: Signal Mnemonics of 20-pin Plug Connector *84*
Table 25: Supported Pins of 20-pin Plug Pin Receptacle Connector on Shuttle *85*
Table 26: Signal Mnemonics of 20-pin Plug Connector *85*
Table 27: Pin Assignment of the USB Type A Connector *86*
Table 28: Pin Assignment of 8-pin RJ45 RS232 Connector (COM1) *86*
Table 29: Pin Assignment of CompactPCI J1 (110-pin type "B" modified) *87*
Table 30: Pin Assignment of CompactPCI J2 (110-pin type "B" modified) *88*
Table 31: Signal Mnemonics of CompactPCI J2 ..... *88*
Table 32: Pin Assignment of CompactPCI J3 (95-pin type "B" modified) *89*
Table 33: Signal Mnemonics of CompactPCI J3 ..... *89*
Table 34: Pin Assignment of CompactPCI J5 (110-pin type "B" modified) *90*
Table 35: Signal Mnemonics of CompactPCI J5 ..... *90*
Table 36: Supported pins of the Parallel ATA 44-pin plug connector on Shuttle *91*
Table 37: Signal Mnemonics of the Parallel ATA 44-pin Plug Connector on Shuttle *91*
Table 38: Supported pins of Serial ATA 22-pin plug connector on Shuttle *92*
Table 39: Signal Mnemonics of the Serial ATA 22-pin Plug Connector on Shuttle *92*
Table 40: Supported Pins of 20-pin Plug Pin Receptacle Connector on Shuttle *93*
Table 41: Signal Mnemonics of 20-pin Plug Connector *93*
Table 42: CPU Address Map *95*
Table 43: PCI I/O Address Map *96*

1 Overview

1.1 Preface

This manual provides general information and instructions on the use of the RAIDStor Module. This includes hardware preparation, installation instructions, operating instructions, configurations, a functional description of the product, environmental ratings and mechanical specifications.

Parts	Description
6210 RAIDStor 6210SH RAIDStor Drive Shuttle Hard Drive Expansion (Optional) DVD Expansion (Optional)	RAIDStor is a network attached storage solution (NAS) representing the leading edge in highly available (HA) embedded systems designed for maximum data security, preservation and integrity. RAIDStor is a leap in technology over conventional box level RAID solutions. RAIDStor is a building block that eliminates difficult file synchronization issues required when implementing HA systems.
6211 RAIDStor 6211SH RAIDStor SATA Shuttle Hard Drive Expansion (Optional) DVD Expansion (Optional)	The RAIDStor Model 6211 utilizes a 6211SH SATA Shuttle, which accepts industry standard 2.5" Serial ATA hard drives. This conversion or upgrade of the RAIDStor Model 6210, replaces the 6210SH PATA Shuttle which only supported industry standard 2.5" Parallel ATA hard drives. The Hard Drive Interface is the only difference between RAIDStor Models 6210 and 6211.

1.2 Introduction

The RAIDStor provides the user with a board level Network Attached RAID solution in a proven 6U x 160 mm industrial form factor. It is hot swappable at the drive and blade level, and designed to implement highly available solutions. Each RAIDStor provides two cPSB compatible Gigabit Ethernet RJ45 ports to enable connections to two different networks. When used in a dual-star cPSB compatible backplane, an Ethernet connection is made to two fabric switches allowing redundant Ethernet connections to be made to other node boards within the cPSB chassis.

The RAIDStor is a hot-swappable 6U PICMG 2.16 node board. It consists of the baseboard and two shuttles for hard disk drives. The shuttles enable the hot swap of the hard drives. Software is supplied to allow the hard drives to be configured for Raid 0 (stripping) or Raid 1 (mirroring). The RAIDStor also supports the addition of an expansion module that adds two additional shuttles. With the expansion module installed, the software also supports Raid level 5 (block stripping + parity). An optional DVD Expansion board can be added to the configuration.

The RAIDStor's advanced features allow automatic network synchronization of data written to an active RAIDStor operating in a mirrored Network Attached Storage (NAS) configuration. Using redundant NAS controllers, data can be mirrored between RAIDStor blades over the network or within each blade in RAID 1, 5, 0+1, 1+1, or 5+1 configurations. Automatic fail over between RAIDStor blades provides a highly available storage solution achieved by implementing an active/backup architecture across the network. The NAS architecture supports fail over and recovery of an individual drive, a NAS controller, or an Ethernet port.

1.3 Photos

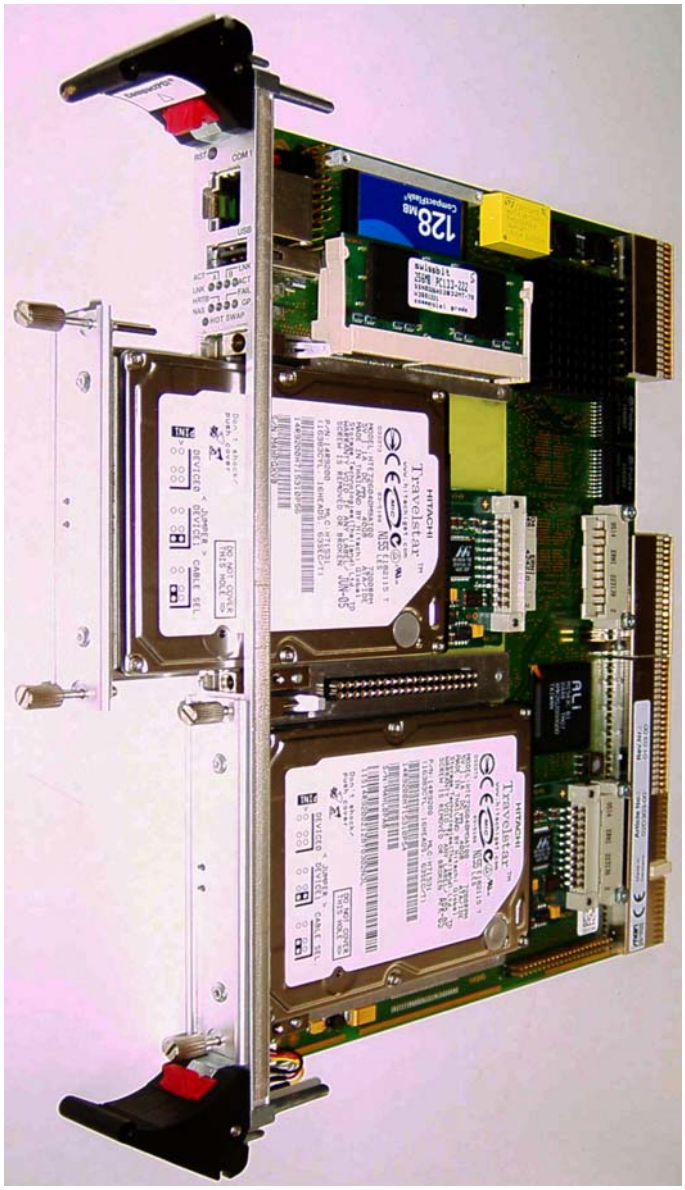


Figure 1: 2 Shuttle 6210 RAIDStor



Figure 2: 4 Shuttle RAIDStor



Figure 3: 4 Shuttle RAIDStor with DVD

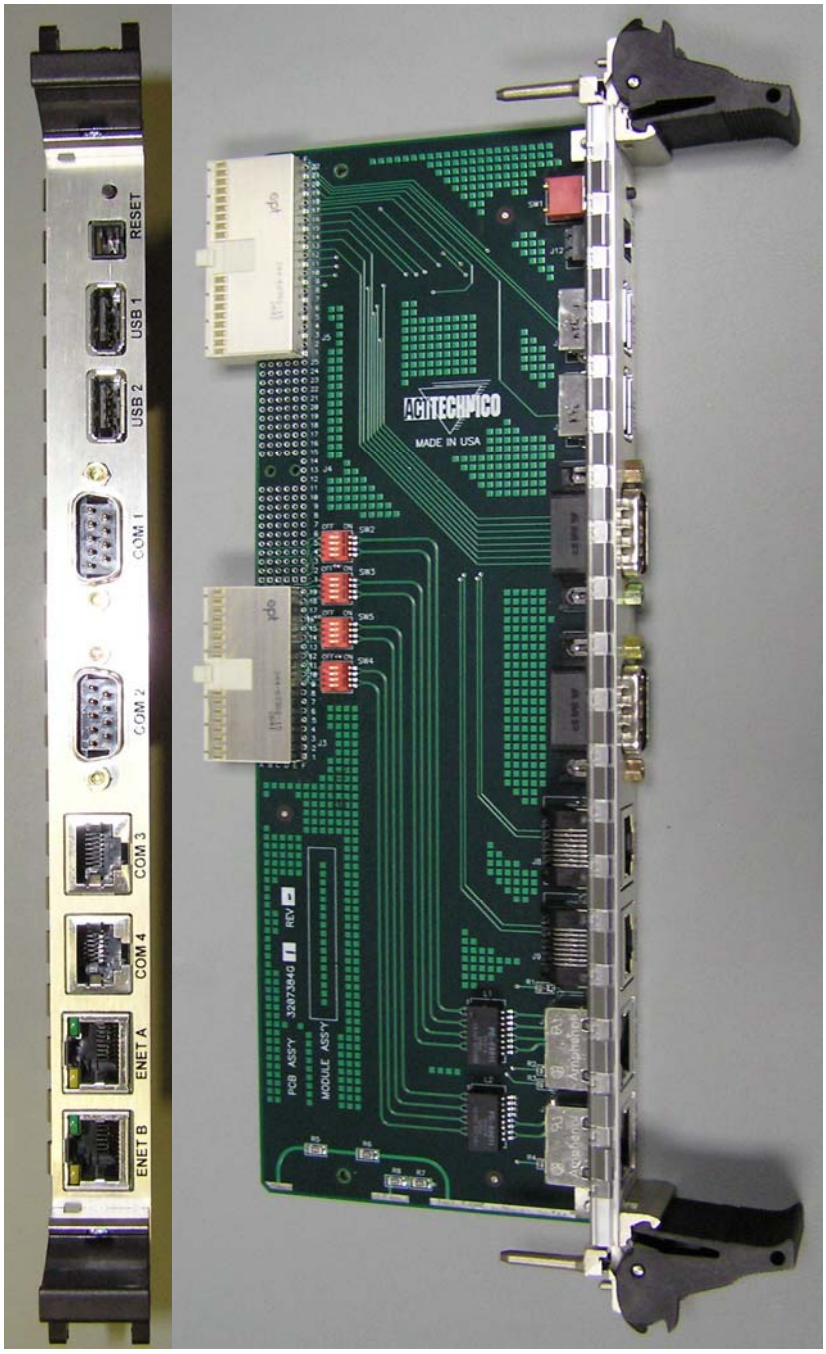


Figure 4: Rear Transition Module

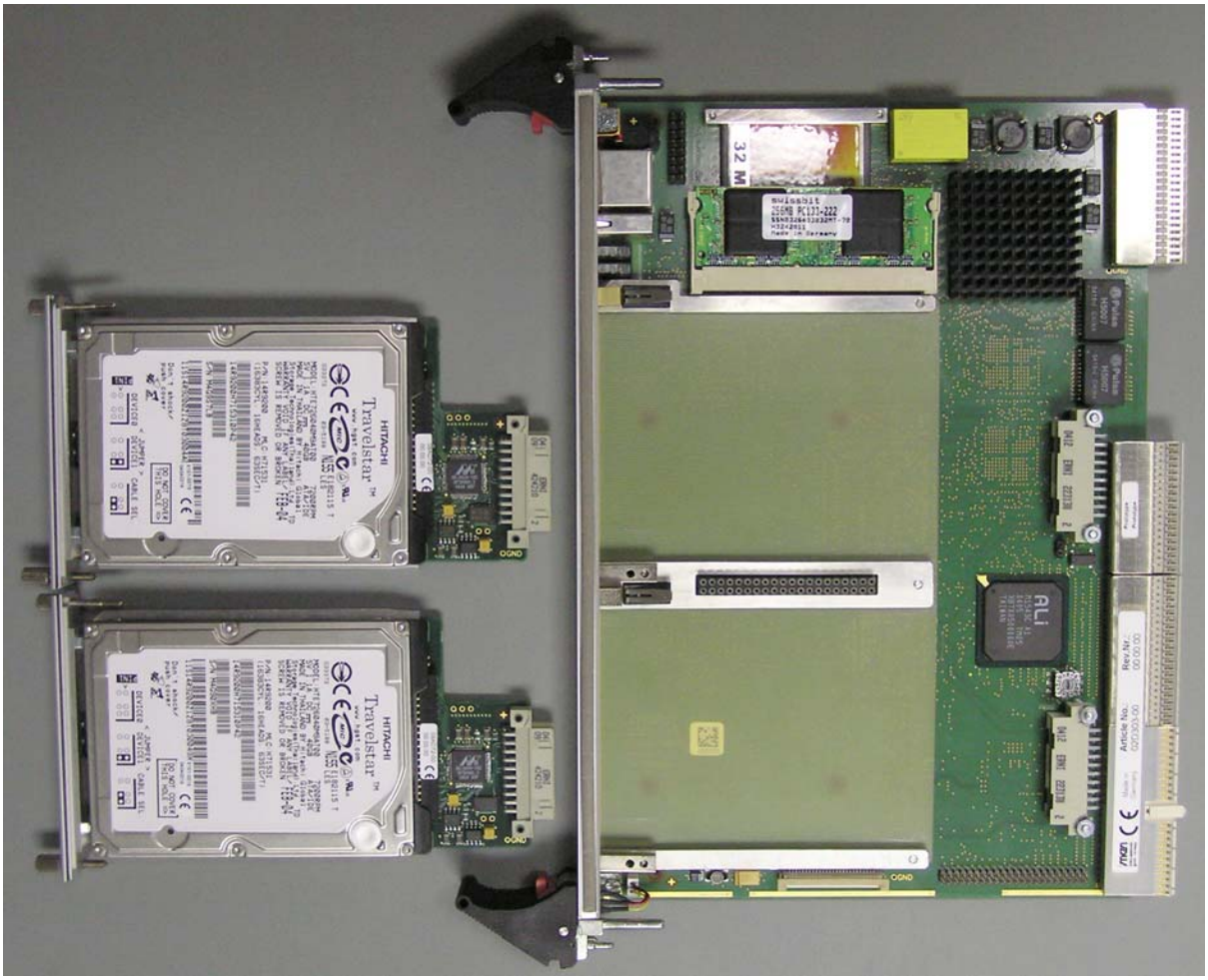


Figure 5: RAIDStor w/ 6210SH shuttles removed



Figure 6: 6210SH Drive Shuttle



Figure 7: RAIDStor with 6211SH Shuttles Removed



Figure 8: 6211SH SATA Shuttle

1.4 Block Diagram and Board Layout

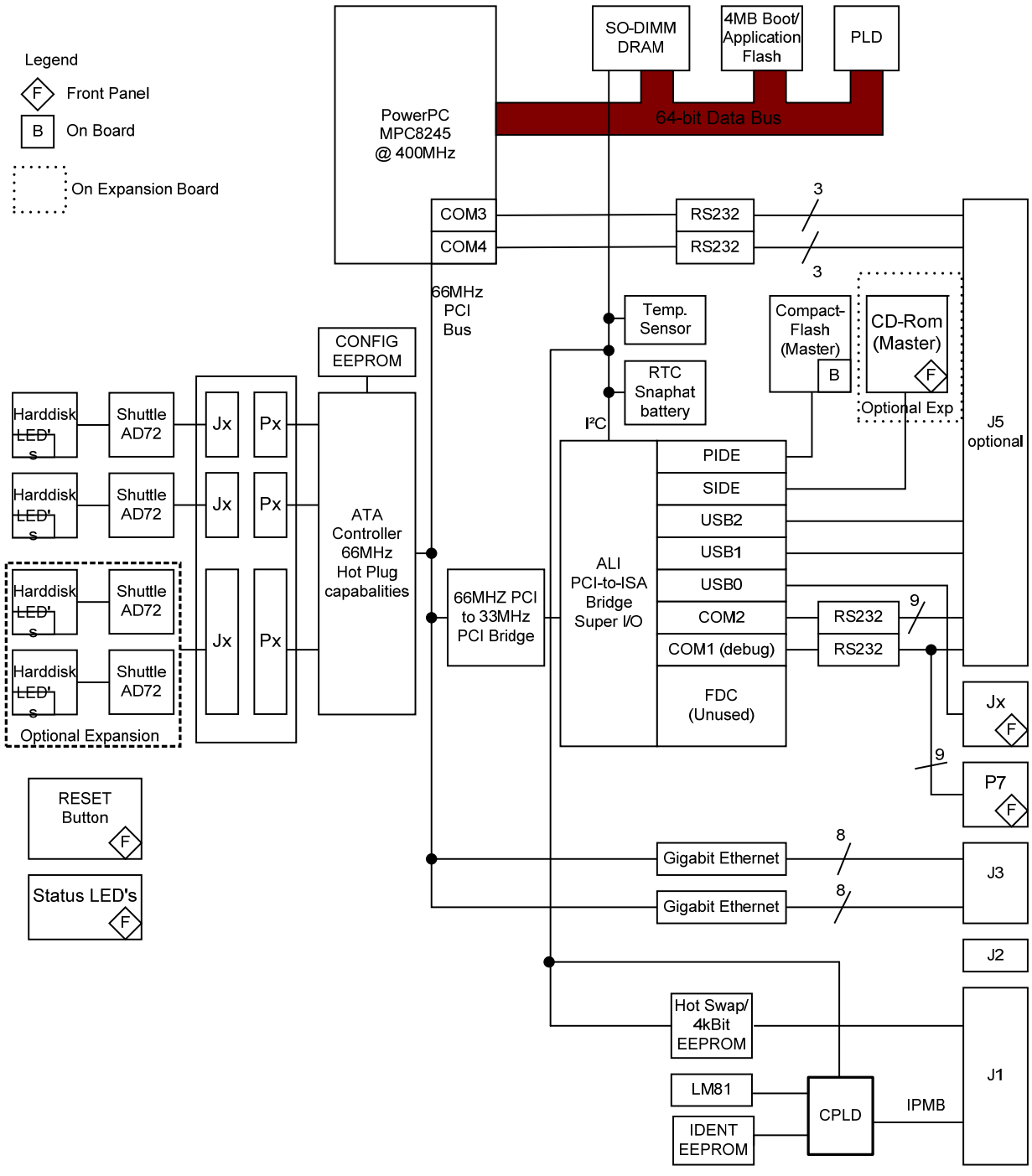


Figure 9: RAIDStor Block Diagram

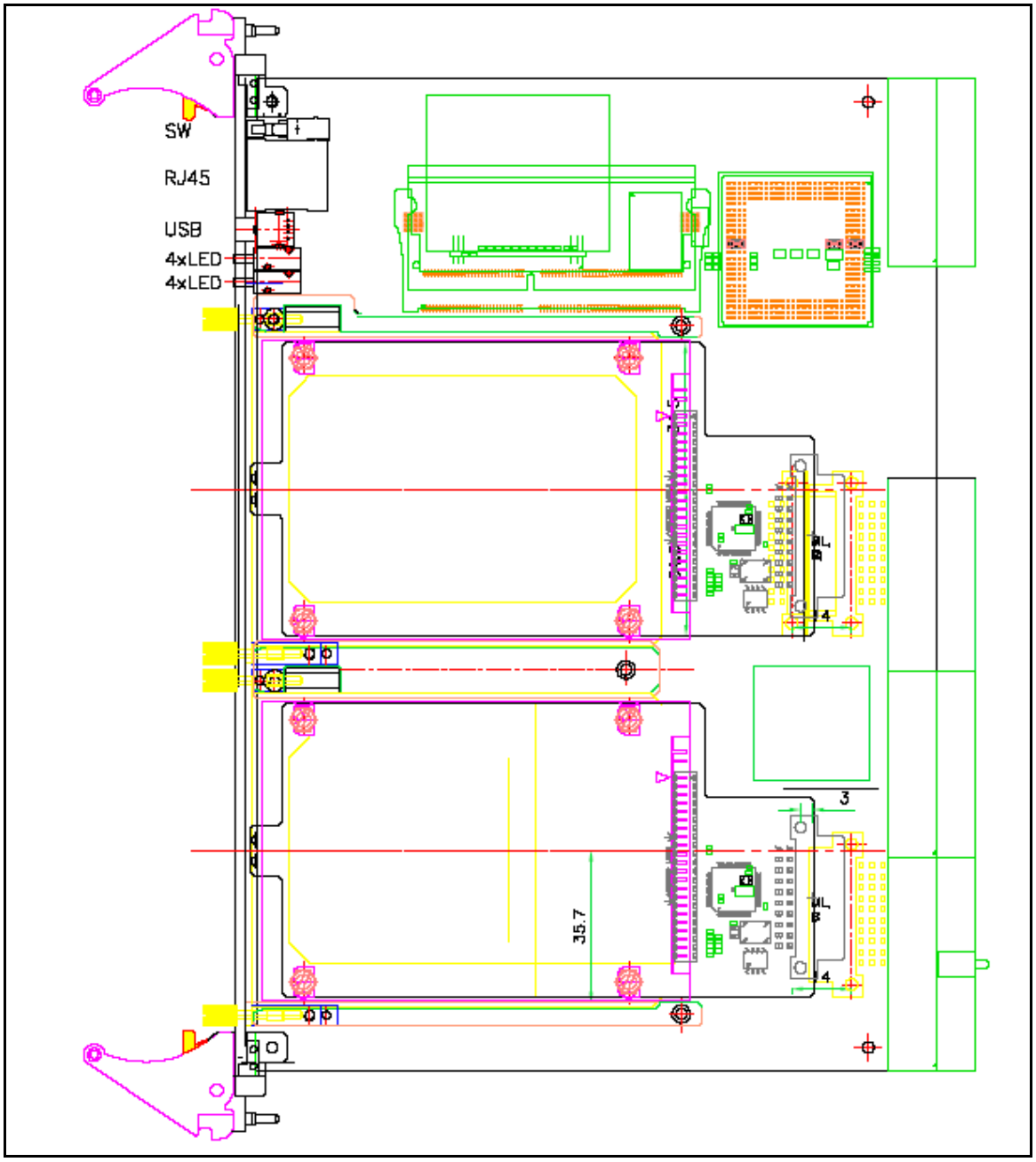


Figure 10: 6210 RAIDStor Hardware Component Layout

1.5 Standards References

Standard	Designation	Edition
PICMG 2.16	Packet Switching Backplane Specification	1.0, 9/5/01
IPMI	Platform Management Bus Communications Protocol Specification	V1.0 November 15, 1999
PICMG 2.1	Hot Swap Specification	R2.0 1/17/00
CF+	CF+ and CompactFlash Specification	Revision 1.4 1999
Serial ATA	High Speed Serialized AT Attachment	Revision 1.0a 7-January-2003
T13/1410D	Information Technology –AT Attachment with Packet Interface – 6 (ATA/ATAPI-6)	Revision 3b 26 February 2002
IEC 60801-2:1991	Electromagnetic compatibility for industrial-process measurement and control equipment, Part 2: Electrostatic discharge requirements	1991 Update
IEC 61000-4-3:2002	Electromagnetic compatibility (EMC) - Part 4: Testing and measurement techniques - Section 3: Radiated, radio-frequency, electromagnetic field immunity test	2002 Update
IEC 60801-4:1988	Electromagnetic compatibility for industrial-process measurement and control equipment, Part 4: Electrical fast transient/burst requirements	1988

Table 01: Standards References

2 General Description

2.1 RAIDStor Hardware Description

The RAIDStor is a 6U hot-swappable PICMG 2.16 node board. It consists of the RAIDStor baseboard and two hard drive shuttles, with the option for HDD and DVD Expansion. The RAIDStor may be plugged into a Node Slot of a PICMG 2.16 Backplane in order to communicate via an Ethernet network with other Node boards. It may also be used in a CompactPCI slot with the addition of the Rear Transition Module (RTM) and external Ethernet cables. There is no CompactPCI bus on this card as it is designed for NAS oepration.

The RAIDStor includes two physical Ethernet ports and supports Hot Swap functionality. The hard disks may be hot swapped from the front panel. LEDs show the status of the hard disks. A USB connector is available on the front panel. A front panel RJ45 connector may be used to connect to the serial console port. LEDs on the front panel provide status information about the board.

2.1.1 Key RAIDStor Hardware Features

CPU

- Motorola PowerPC
  - MPC8245
  - 400MHz

Memory

- Level 1 Cache integrated in MPC8245
- SO-DIMM slot for 256MB SDRAM
- 100MHz memory bus operation
- Flash 4MB
  - 8-bit data bus
- Serial EEPROM 512 bytes for factory settings
- CompactFlash (TM) card interface for Flash ATA (true IDE) via on-board IDE

EEPROM

- 4KBit for factory data, MENMON parameters
- Provided by SMH4042 on board hot swap controller

Interfaces

- Two 10/100/1000 Mbits/s Ethernet channels
  - Intel 82541ER
- Connection via rear I/O J3 connector
- Four UARTs (COM1..4)
  - Accessible via rear I/O J5 connector
  - COM1 also via RJ45 at front panel
  - COM1, COM2 with full control lines
  - COM3, COM4 implemented as three-wire interface
- IDE port
  - For CompactFlash and CD-ROM drive
- USB ports
  - 2 USB 1.1 interfaces on rear I/O J5 connector
  - 1 USB 1.1 at front panel for downloading user software
- Four serial ATA interfaces
  - 2 or 4 shuttle bays for 6210SH or 6211SH shuttles with 2.5" hard disks
  - Only one CompactPCI slot needed even with hard disk shuttles mounted

Local PCI Bus

- PCI Spec. 2.2 compliant
- 32-bit data bus, 66MHz, 3.3V

CompactPCI

- For use in CompactPCI Packet Switching Backplane
- NO connection to CompactPCI backplane bus
- Hot-swap support

Miscellaneous

- Serial real-time clock with integrated 56-byte NVRAM
- Serial hardware watchdog in supervisory circuit
- Temperature sensor
- Hex switch for factory use
- User LEDs (external)

ALI M1543C Super I/O Controller

- 33MHz local PCI frequency
- Connects to the Processor 66MHz local pci bus via a TI 2050 66MHz to 33Mhz bridge
- 32bit data width
- PCI master/target
- Supports PCI interrupt controller
- USB ports
- IDE controllers
- 2 16550 compatible UARTs with 16-byte transmit/receive FIFOs
- The ALI M1543C is on PCI bus 1, it is seen by software as four PCI devices, a PCI-ISA controller, an IDE controller, a PMU and a USB controller.

Hot Swap Controller

- SMH4042
- Local I2C address 0x54
- Host VCC detection
- Card voltage sequencing
- Card insertion detection
- Card voltage monitoring
- Provides a 1V reference for pre-charging the CPCI bus signal pins
- Power on reset

CPLD

- Accessible by local CPU
- Provides Front Panel LED controls
- Provides control for on board reset logic
- Provides read access to slot geographical addresses
- Provides watchdog toggle bit to Watchdog in the Real Time Clock
- Provides reset and power control for shuttles
- Provides interrupts and interrupt masking for disk shuttles
- Monitors the board ejector switch
- Provides read access to hex switch on baseboard
- Provides test access to the IPMB from the local I2C bus. This function is not intended for customer use.

Software Support

- Linux
- MENMON

2.2 RAIDStor Software Description

The RAIDStor runs the ElinOS 3.1 Linux 2.4.25 PowerPC distribution from Sysgo AG in Germany. Currently over 70 additional software packages are installed on the RAIDStor. Some of these packages are new and were not included in the original ElinOS distribution. Others are updates to existing packages. Some packages are directly related to the needs of the High-Availability (HA) software and others add useful functions not related to HA.

The RAIDStor includes a complete GNU based command line software development environment.

2.2.1 Key RAIDStor Software Features

Feature	Description
Operating System	• ElinOS 3.1 distribution – Linux 2.4.25 kernel
Software Disk RAID	• Provided by the Linux 'md' driver. • RAID-0, RAID-1, and RAID-5 supported. This includes RAID mode combinations, such as RAID-0+1.
Network RAID	• Provided by the Distributed Replicated Block Device (DRBD) module. • Provides RAID-1 (mirroring) support between two blades over network. • Only available in dual blade systems.
Blade Failover	• Provided by the 'heartbeat' software package. • Monitor hosts and informs the cluster when one goes down, controls starting/stopping services on the two RAIDStor blades. • Only available in dual blade systems.
Network Interface Failover	• Provided by the Linux 'bonding' driver. • Ties together multiple physical network ports so that they provide redundancy and automatic port failover.
Basic Protocols and Services	• Provided by numerous packages, including: Linux OS, FSF packages, and other GPL packages. • All of these are provided by free software packages.
Web/SNMP Configuration Interface	• Provided by Webmin modified modules and some custom code. • The Web/SNMP system API is a combination of Webmin scripts, system scripts, system configuration files, and custom scripts.
Configuration Synchronization	• Provided by the SNMP functionality. • Keeps active and standby blades in sync as well as initializing new blades • Only applicable to dual blade systems.
Shuttle Hot Swap	• Provided by the CPLD and other hardware • ACT/Technico enhancements to Linux • Allows automatic disk rebuild

Table 02: Key RAIDStor Software Features

2.3 Single Blade RAIDStor

The single blade RAIDStor is available with two or four disk shuttles. It can occupy 1 to 3 physical backplane slots. The available RAID levels are dependent on the RAIDStor configuration purchased. Table 3 below lists all of the available RAID level configurations for a single blade RAIDStor.

Single/Dual	# Shuttles	RAID
Single	2	0
Single	2	1
Single	4	5
Single	4	0

Table 03: Single Blade RAID Levels

The RAIDStor has dual network interfaces that are *bonded* to appear as one. In this case, *bonded* means that both physical Ethernet ports have the same IP and MAC addresses. A special Linux network driver is used to handle the bonded Ethernet ports. If a network or interface fails, then the driver automatically fails over to the secondary port. Once the primary port or network is repaired, then it automatically takes over from the secondary port.

A single blade RAIDStor includes 2 or 4 disk drives mounted on removable shuttles. Disk redundancy is provided by RAID-1 (mirroring) across all partitions on the 2 disk version and RAID-5 (stripped, parity) on the 4 disk version. In either configuration, the system can lose one disk drive without affecting data integrity. A single blade RAIDStor configured for RAID-0 is effectively non-functional if a single drive fails.

Software RAID is used to connect the disk drives so that they appear as a single drive with multiple partitions. Some of the partitions are used for internal blade operations, such as the Root File System and a backup, and one is for external use. The external partition is exported via FTP, NFS, and SAMBA (SMB/CIFS) protocols.

When a shuttle is removed, the system will remove its partitions from their associated RAID partitions. RAID-1 and RAID-5 configurations will continue to function with a single missing disk drive. A RAID-0 configuration will not operate with a missing drive. When a shuttle is installed, the partitions on the new shuttle are compared with the RAID partitions; if they are different, the shuttle will be repartitioned so that it is the same. The shuttle's partitions will then be added and resynchronized with the RAID partitions. This happens without user intervention. **Warning, installing the wrong drive shuttle into an empty shuttle bay will cause the hard drive to be overwritten.**

The system will not allow removal of more than one shuttle at a time and will ignore requests to remove a second shuttle. It will also ignore requests to remove a shuttle during any type of RAID resynchronization.

Optionally, the external partition can be configured as RAID-0 (stripped), this mode offers higher capacity, at the expense of reduced reliability. If a shuttle is removed, the external partition is shutdown and no longer served. If the same shuttle is replaced, then the external partition is restarted with no loss of data. However, the failure or replacement (with a different drive) of any shuttle will cause the loss of all user data on the external partition. If data reliability is also needed, it is recommended to use a single blade RAIDStor configured for RAID-1 or RAID-5, or a dual blade RAIDStor.

2.4 Dual Blade RAIDStor

The dual blade RAIDStor provides all the features of the single blade RAIDStor. It additionally provides dual CPU redundancy. The dual blade RAIDStor provides data mirroring over the network across the blades. If a blade fails, then the backup blade automatically takes over all network and disk services that were running on the failed blade with no loss of data to the user. When the failed blade is replaced, it is resynchronized with the backup blade and then takes over from the backup blade. Since the dual blade RAIDStor uses the network to communicate between the blades, the blades do not have to be physically located near each other. In fact, they could be in different chassis, different rooms, different buildings, different states, or different countries, as long as they can share a network connection.

The dual blade RAIDStor provides a blade monitoring system called *heartbeat* and a data mirroring system called the *Distributed Replicated Block Device* (DRBD). Heartbeat is used on both blades to continuously monitor the status of the blades. If the primary

blade stops, then heartbeat will detect it and initiate a fail over to the backup blade. The backup blade will then take over from the primary blade serving the data on the external partition.

The DRBD software is used to mirror the data on the external partition from the primary blade to the backup blade. It ensures that the backup blade is always in sync with the primary blade so that it can take over from the primary blade at any time. There is a short transition delay (about 30 seconds) during failover. However, the user will see no loss of data during this time, even if the primary blade is stopped abruptly (or crashes). Any transaction in progress during the failover will be temporarily paused and resumed once the failover is complete. Once the failed blade is restored, DRBD will resynchronize the data across the blades and heartbeat will bring it back online. The system continues to function normally even during data resynchronization.

The dual blade RAIDStor offers the same RAID levels as the single blade RAIDStor. The external partition can be setup as RAID-0 or RAID-5 if four drive shuttles are present. The internal RAID partitions are setup as RAID-1 or RAID-5. The following table lists the possible configurations of a Dual Blade RAIDStor.

Single/Dual	# Shuttles	RAID
Dual	2	1+1
Dual	2	1+0
Dual	4	1+0
Dual	4	1+5

Table 04: Dual Blade RAID Levels

Loss or removal of a shuttle on the primary blade will automatically force a failover to the backup blade. Reinsertion of the shuttle will first resynchronize the RAID-1 or RAID-5 partitions, then the RAID-0 partition will be verified and rebuilt if it is bad, then DRBD will resynchronize external partition with the backup blade, finally heartbeat will initiate a fail back to the primary blade. There should be no loss of user data at anytime and the system continues to function normally even during data resynchronization.

2.5 Front Panel LEDs

There are nine LEDs on the front panel of the RAIDStor.

- Two LEDs indicate Ethernet Link status
- Two LEDs indicate Ethernet Activity
- The standard Blue Hot Swap LED is used to indicate that the board is ready for removal
- The remaining four LEDs are software controlled and used to indicate the current status of the RAIDStor blade. The silkscreen on the front of the RAIDStor identifies these LEDs by function. These four LEDs blink twice on power up. If the Red LED labeled *Fail* does not blink on power up then IPMB (IPMI) power is not available in the slot currently occupied by the RAIDStor. The usage of the LEDs is as follows.
- NAS LED – This LED is illuminated by software if the RAIDStor is currently the ‘primary’ Network Attached Storage blade and is currently exporting a file system.
- HRTB LED – This LED blinks to indicate that the heartbeat application is currently running on this blade. In normal operation on a dual blade RAIDStor it will blink on both blades. It is not used a single blade system.
- Fail LED – This LED is illuminated by software to indicate a critical failure.
- GP LED – This LED is illuminated by software to indicate that a drive is currently being re-built (sync-ed) by the software.

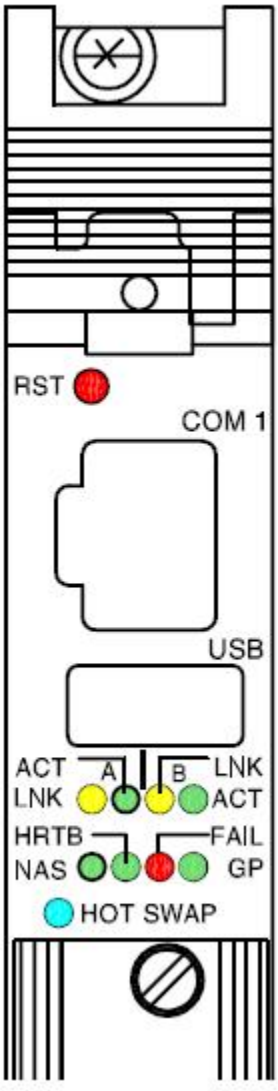


Figure 11: Front Panel LEDs

On a dual blade RAIDStor, the HRTB LED will be blinking on both blades and the primary (active) blade will also illuminate the NAS LED. On a single blade RAIDStor, the HRTB LED will not be illuminated and the NAS LED will be illuminated if it is serving the external partition.

2.5.1 Drive LEDS

There are 2 LEDS on each disk drive shuttle. The top yellow LED is illuminated anytime the disk is accessed. It is not controlled by software. The lower green LED is software controlled and is used to indicate drive status with various blink patterns as follows:

- Green LED off – drive is offline and may be removed.
- Green LED on – drive is online and may NOT be removed w/o initiating a removal.
- Green LED blinking rapidly – drive is in the process of being removed or installed, wait until done.
- LEDs on each shuttle are toggling between the upper and lower shuttle – the system is performing a RAID 1 resynchronization, wait until done. (A 4 shuttle system performing RAID-5 resynchronization will toggle the drive LEDS in a circular manner). The rate of toggling is used to indicate the remaining time. Slow toggling indicates that the process is just beginning. Fast toggling indicates that it is nearing completion.

- LEDs on each shuttle all toggle simultaneously on each blade – the system is performing a blade-to-blade resynchronization, wait until done. The rate of toggling is used to indicate the remaining time. Slow toggling indicates that the process is just beginning. Fast toggling indicates that it is nearing completion.

2.6 Operating System

The RAIDStor ships with an Elinos 3.1 distribution of Linux 2.4.25 preloaded and configured. Linux is configured to provide RAID services across the two and four hard drives and across the two RAIDStor blades in a dual blade configuration.

2.7 Compact Flash

The Compact Flash normally ships with a single file installed on it named pImage. This is the Linux Kernel, which is booted by the RAIDStor. The root filesystem resides on the hard drives mounted on the disk shuttles. This is device /dev/md/0. A backup root file system resides on /dev/md/1

2.8 Required Linux Modules

The following Linux modules are required and included for proper RAIDStor operation.

- *cpld* - The cpld module provides interrupt service and control for the disk shuttles.
- *men_sysparam*- The mem_sysparam module provides kernel user space services.
- *Distributed Replicated Block Device* (DRBD) - This module provides mirrored RAID across the RAIDStor blades in a dual blade configuration.
- *bond* - The standard Linux bonding driver is used to provide Ethernet failover services between the two Gigabit Ethernet ports on the RAIDStor.
- *i2c-sms24* - This module provides access to devices on the internal i2c bus.

2.9 Web Interface

The RAIDStor provides a Web Interface for control and status. On a single blade RAIDStor, it is accessed at the board IP address; which is http://192.168.0.201 by default. On a dual blade RAIDStor use the NAS address; which is http://192.168.0.200/ by default. The default userid is **admin** with a password of **admin1423**.

2.10 SNMP Interface

The RAIDStor can be managed via SNMP or the web interface. The web interface actually uses SNMP to interrogate and control the RAIDStor. The MIB files can be found in /usr/share/snmp/mibs/. Two MIB files specific to the RAIDStor are provided.

- ACTTECHNICO-CORP-MIB.txt
- ACTTECHNICO-RAIDStor-MIB.txt

3 RAIDStor Usage and Operation

The RAIDStor is designed to be installed in a PICMG 2.16 node slot. Be sure to observe proper anti-static procedures when installing the blade in a chassis. For proper operation, the RAIDStor needs to be plugged into a cPCI Rack that supports hotswapping and IPMBus power for the IPMB Section of the board.

Note, without IPMB power, the RAIDStor powers up, but the fail LED will not blink two times like all other user LEDs.

3.1 Initial Installation



The RAIDStor is shipped preloaded and configured to auto-boot. Install the blade or blades into an appropriate chassis. If a dual blade RAIDStor is to be deployed in multiple chassis', then a network connection between the two chassis is needed to allow the Network RAID to function between the blades.

On power up, the 4 user LEDs on the front panel of the RAIDStor will blink twice. If you wish to interrupt the boot sequence connect a serial console to the front panel of the RAIDStor and press the <**escape**> key. On a dual blade RAIDStor, the system is booted when the HRTB (heartbeat) LED blinks on both blades and the NAS (network attached storage) LED is illuminated on the primary blade. On a single blade RAIDStor, the system is booted with the NAS (network attached storage) LED is illuminated. The NAS LED indicates that the blade is currently actively exporting the user file system.

If the GP (drive sync) LED is illuminated on either blade, a drive is in the process of being resynchronized.

Important Note, in a dual blade configuration, both blades should normally be started and shutdown at the same time. If the blades are not started and stopped together, then at least one blade will require a short or long resynchronization. The blades are currently configured to wait 2 minutes after booting for their partner blade to bootup. If, after 2 minutes, the partner blade is still not present, then the blade will automatically take control of the NAS address and come on-line. Once the partner blade has booted, it will automatically connect and resync its data with the other blade.

3.2 Shuttle Removal/Insertion Process

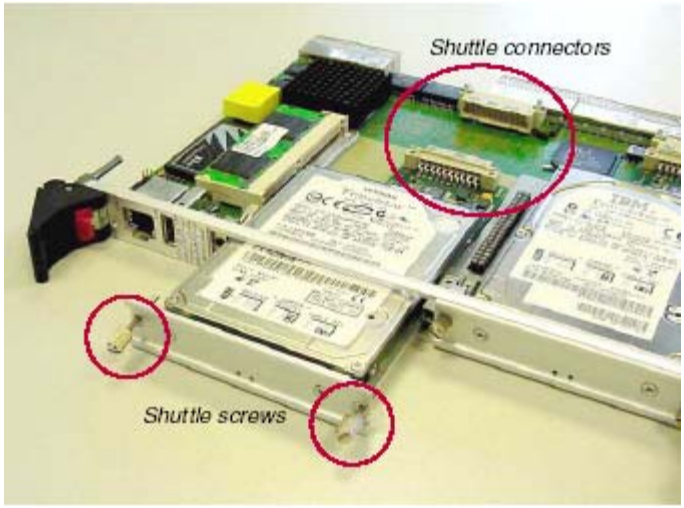


Figure 12: Shuttle Insertion/Removal

There are two thumbscrews holding each of the hard drive shuttles. The upper thumbscrew (vertical mounting) or leftmost (horizontal mounting) is connected to a micro-switch on the base of the RAIDStor that generates an interrupt when it opens or closes. This interrupt causes the software to flush buffers, shutdown the drive, and turn the Green LED off to indicate it is OK to remove the drive shuttle. ***It is important*** to let the software finish a drive insertion and/or removal before starting a new operation.

Remember in a RAID0 configuration a blade is non-functional with a drive removed. In any other RAID configuration only 1 shuttle can be removed at a time.



The shuttle removal sequence is as follows:

- Wait until all green drive LEDs have stopped blinking, verify that all are illuminated. If any are blinking or not illuminated then shuttle removal will be ignored.
- Unscrew upper thumbscrew all the way out.
- Wait for 5-10 seconds for the green drive LED to start blinking, indicating it is processing the shuttle removal. If it does not start blinking, then there is some reason that the shuttle cannot be removed.
- After about 30 seconds the green drive LED is turned off, indicating it is safe to remove the shuttle. If it does not turn off, then there is some reason that the shuttle cannot be removed.
- Unscrew the lower thumbscrew and remove the shuttle.
- ***You will not be able remove*** any other drive shuttles on either blade until the first drive you removed has been returned to service. That is indicated to user by the failure of the shuttle's green LED to go out in response to removing the shuttle screw.

The shuttle insertion sequence is as follows:

- Insert the shuttle and screw in the upper and lower thumbscrews.
- Wait for 5-10 seconds for the green drive LED to start blinking, indicating it is processing the shuttle insertion. If it does start blinking, then there is some reason that the shuttle cannot be inserted. Due to the mechanical nature of the shuttle switch it is necessary to de-bounce the switch in the interrupt driver, and again in the PERL script that processes the shuttle insertion. It is possible for the script to incorrectly think a removal is being requested on a shuttle install. If a serial console is attached to the RAIDStor this will be apparent by the messages on the console. If no console is attached and a drive does not power up, back out the thumbscrew and then re-insert it. If the drive does not power up on the second attempt something is wrong.
- The Yellow LED and Green LED will remain blinking while the hard drive re-syncs.
- After a potentially long period the green drive LED is turned ON solid, indicating that the shuttle is online and resynchronized. This may only take a few minutes depending on configuration. A lengthy delay can be caused by resynchronizing the data across the drives (and blades in a dual blade system). If the LED does not turn on, then there is some reason that the shuttle cannot be inserted. There should be no interruption of service while the drives are being resynchronized.
- ***You will not be able remove*** any other drive shuttles on either blade until resynchronization has completed and the drive has returned to service. That is indicated to the user by the failure of the shuttle's green LED to go out in response to removing the shuttle screw.

3.2.1 Dual Blade RAIDStor Failover Policy

The current active/standby blade failover policy is as follows.

- The designated primary blade always exports the shared file system when it has current data.
- The designated backup blade takes control when it determines the primary blade has failed.
- The primary blade will take always control back from the backup blade when it returns to service.

3.3 RAIDStor Shutdown

The easiest way to shutdown the RAIDStor is to depress the Ejector switches. On a dual blade RAIDStor, typically do both blades at the same time.

To remotely shut down the RAIDStor blade(s) on a single or dual blade RAIDStor, log onto the administration account via the web interface by pointing your browser at the NAS address.

- http://192.168.0.200 (put your NAS address here)
- Login ID = admin
- Password = admin1423
- Select *System*
- Press *Reboot blade* or *Halt blade* (Single Blade RAIDStor)
- Press *Reboot both blades* or *Halt both blades* (Dual Blade RAIDStor)

To shut down a single blade on a dual blade RAIDStor:

- Select *Blade Info*
- Using the blade pull down menu, select either *Active Blade* or *Standby Blade*
- Pres *Reboot Blade* or *Halt Blade*

In a dual blade RAIDStor, halting the *Active Blade* from this screen will cause a failover to the *Standby Blade*. Stopping Heartbeat from this screen will also cause a failover to occur.

Blade shutdown typically takes about 30 – 60 seconds. The blade may be removed when the blue HOT SWAP LED is illuminated. Do not remove the blade until the blue HOT SWAP LED is illuminated.

It is also possible to shutdown or reboot a RAIDStor with the Linux "snmpset" command that is part of the open source net-snmp package. For example, assuming that you want to reboot or halt a RAIDStor at IP address 10.0.0.99 the following command(s) could be used:

- *snmpset –v2c –c public –mALL 10.0.0.99 rsRaidstorReset.0 i 1* (reboot blade)
- *snmpset –v2c –c public –mALL 10.0.0.99 rsRaidstorReset.0 i 2* (halt blade)

3.4 Default Passwords

The Linux **root** login password is set to **root1423**.

The web administration account uses a login ID of **admin** with the password set to **admin1423**.

The web administrator is not a normal Linux user account.

3.5 Default IP Address Assignments

The RAIDStor is shipped with the following default IP addresses.

3.5.1 Single Blade RAIDStor

The default IP address assignments for a single blade RAIDStor is follows.

- Primary 192.168.0.201

3.5.2 Dual Blade RAIDStor

The default IP address assignments for a dual blade RAIDStor is follows.

- NAS 192.168.0.200 (Shared Network Address)
- Primary 192.168.0.201
- Backup 192.168.0.202

The Primary and Backup IP addresses are internal addresses that only need to be changed if they would cause a problem in the network they are being installed into. One or more public addresses are shared between the blades and exported to the world for the purpose of making the network-attached storage available to users.

The Primary and Backup IP addresses exist permanently on the blades they were assigned. The NAS address is dynamically switched to the RAIDStor blade currently providing a network file system.

3.5.3 Changing the IP Address

The three options to change the IP addresses of a RAIDStor blade are using the webpage, issuing an snmp command call, or editing the /etc/hosts file.

3.5.3.1 Via Webpage

To remotely change the IP address on a single or dual blade RAIDStor, log onto the administration account via the web interface by pointing your browser at the NAS address.

- http://192.168.0.200 (put your NAS address here)
- Login ID = admin
- Password = admin1423
- Select *Network Settings*

Modify the settings and press the *Submit* button after you have made your changes. **Note** once the IP address change is made the user will need to change their web connection accordingly to access the new IP address.

3.5.3.2 Via SNMP

The IP address of a RAIDStor can also be changed via SNMP with the "snmpset" command that is part of the open source net-snmp package. For example, assuming that you want to change the IP address of a RAIDStor from 192.168.0.19 to 192.168.0.20 the following command could be used:

- *snmpset –v2c –c public –mALL 192.168.0.19 rsNetworkVirtualIPAddress.0 a 192.168.0.20*

3.5.3.3 Via /etc/hosts

A 3rd option to changing the IP addresses is by editing the /etc/hosts file, which is explained in Section 5.3.

3.6 RAID Partitioning

The RAID driver combines several disk partitions to make them appear as a single disk with RAID capabilities. Once setup, a small data field called the *persistent super block* is stored on each participating disk. This hidden field is read by the operating system during boot up to setup the RAIDs.

3.6.1 Root File System RAID Partitions

By default, there are 2 Root File System (RFS) partitions. One partition is used as the Linux RFS, the other is an emergency backup copy. These partitions use RAID-1 (mirrored) across the 2 drives on each blade when a disk expansion board is not present. RAID-5 is used when there are 4 drives installed on a single blade. They are unique and local to each blade. They are not mirrored across blades. Also, the primary RFS is not mirrored to the backup RFS so that a problem will not be copied between partitions.

The location of the RFS is stored in an EEPROM and passed to the Linux kernel on boot. If the primary RFS partition becomes corrupted, then the blade could be booted from the backup RFS. To boot from the backup copy of the RFS (/dev/md/1) you must update the EEPROM with the *ee-kerpar* command from the RAIDStor firmware. However, if any RAIDStor service settings were changed, then the backup will be out of sync and will not be compatible with the other blade.

It is not recommended that any RFS partition be changed. Changing those partitions improperly could cause a system failure. Also, you cannot change the partition of the active RFS and to do so you must boot from an external NFS served RFS. Changes to the booting configuration require a serial terminal connection to the RAIDStor. If the RFS partition is changed, then all operating system files need to be restored. Also, all RAIDStor service settings will be lost and will also need to be restored.

3.6.2 Single Blade RAIDStor Partitions

A single blade RAIDStor is typically setup with 5 RAID partitions as follows:

Partition Type	Partition Name	RAID device file	Type	Size
RFS Partitions	Root File System primary (RFS)	/dev/md/0	RAID-1 (mirrored)	1.0 Gbyte
	Root File System backup	/dev/md/1	RAID-1 (mirrored)	1.0 Gbyte
Unused	None	/dev/md/2	RAID-1 (mirrored)	1.0 Gbyte
User Partitions	private	/dev/md/3	RAID-0 (striped) or RAID-1 (mirrored)	128 Kbyte
	raidstor	/dev/md/4	RAID-0 (striped) or RAID-1 (mirrored)	varies

Table 05: Typical RAID Partitions 2 Drives Single Blade

Partition Type	Partition Name	RAID device file	Type	Size
RFS Partitions	Root File System primary (RFS)	/dev/md/0	RAID-5 (parity)	1.0 Gbyte
	Root File System backup	/dev/md/1	RAID-5 (parity)	1.0 Gbyte
Used	None	/dev/md/2	RAID-5 (parity)	1.0 Gbyte
User Partitions	private	/dev/md/3	RAID-0 (striped) or RAID-5 (parity)	128 Kbyte
	raidstor	/dev/md/4	RAID-0 (striped) or RAID-5 (parity)	varies

Table 06: Typical RAID Partitions 4 Drives Single Blade

3.6.3 Dual Blade RAIDStor Partitions

On a dual Blade RAIDStor, by default, there are 3 DRBD partitions. One partition is for DRBD internal use only and is used to store local (i.e., per blade) DRBD metadata. Metadata are changes made to a DRBD partition, but not yet reflected on the partner blade. DRBD uses the Metadata to quickly resynchronize DRBD disks. The use of a metadata partition means that a crashed blade does not need to resynchronize every sector upon boot up, but can instead compare its metadata with the partner's metadata to determine which sectors need resynchronizing. This greatly improves the resynchronizing performance by 2 or more orders of magnitude. There is one drawback, during board replacement, the metadata partition must be initialized by performing a lengthy sector-by-sector synchronization between blades. On a system with 60 GB hard drives this process will take many (up to 8) hours.

One partition is used for private shared data. This is state information for NFS and SAMBA that needs to survive a failover .

There is typically one exported user partition, but there could be more (depending on the customer's needs and requirements).

A dual blade RAIDStor is typically setup with 5 RAID partitions as follows:

Partition Type	Partition Name	RAID device file	Type	Size
RFS Partitions	Root File System primary (RFS)	/dev/md/0	RAID-1 (mirrored)	1.0 Gbyte
	Root File System backup	/dev/md/1	RAID-1 (mirrored)	1.0 Gbyte
DRBD Metadata Partition	None	/dev/md/2	RAID-1 (mirrored)	1.0 Gbyte
DRBD Partitions	private	/dev/drbd/0	RAID-0 (striped) or RAID-1 (mirrored)	128 Kbyte
	raidstor	/dev/drbd/1	RAID-0 (striped) or RAID-1 (mirrored)	varies

Table 07: Typical RAID Partitions 2 Drives Dual Blade

Partition Type	Partition Name	RAID device file	Type	Size
RFS Partitions	Root File System primary (RFS)	/dev/md/0	RAID-5 (parity)	1.0 Gbyte
	Root File System backup	/dev/md/1	RAID-5 (parity)	1.0 Gbyte
DRBD Metadata Partition	None	/dev/md/2	RAID-5 (parity)	1.0 Gbyte
DRBD Partitions	private	/dev/drbd/0	RAID-0 (striped) or RAID-5 (parity)	128 Kbyte
	raidstor	/dev/drbd/1	RAID-0 (striped) or RAID-5 (parity)	varies

Table 08: Typical RAID Partitions 4 Drives Dual Blade

3.7 Networking

The RAIDStor is equipped with two 10/100/1000 Mbits/s Ethernet channels controlled by an Intel 82541ER. These Ethernet ports are routed to the PICMG 2.16 connector on the backplane. The rear transition module also picks up these signals.

3.7.1 Bonding Driver

The bonding driver is used to tie together the two physical Ethernet network ports so that they provide redundancy and automatic port failover. The bonding driver has four different modes. The RAIDStor uses *Active Backup Mode*, where only one network port is active at a time. If the primary port fails, the bonding driver automatically fails over to the backup port. The configuration file is actually the script: ***/etc/init.d/bonding.sh***.

3.7.1.1 Network Interface Options

The bonding driver picks up its network address from ***/etc/hosts*** entry labeled *myraidstor*. It also creates a route from the bonding interface to the network address from */etc/hosts* entry labeled *gateway*.

Optionally, it can pick up a *netmask* and *broadcast* address from ***/etc/hosts***. Those options can be useful if the default network derived values are not correct.

To add a netmask, add the IP alias, *netmask*, to ***/etc/hosts***. To add a broadcast address, add the IP alias, *broadcast*, to ***/etc/hosts***

3.7.1.2 Bonding Script

Starting the bonding driver is a multi-step process that is started with the script, ***/etc/init.d/bonding.sh***. The steps are as follows:

1. The bonding driver is a module that must be loaded into Linux; it is not linked to the Linux kernel. The mode of operation and various parameters are specified on the command line when the bonding driver is loaded.
2. The bonding driver network interface is initialized and its IP address is set using the *myraidstor* alias defined in ***/etc/hosts***.
3. The two Ethernet drivers, *eth0* and *eth1* are made into slaves under control of the bonding driver.
4. The default route to the gateway is added to the routing table.

Each step and the options available will be discussed in detail in the following sections.

3.7.1.2.1 Loading the Bonding Driver Module

Loading the bonding driver module is the first step when starting the bonding driver. This step consists of loading the bonding driver module, setting its mode of operation, and setting various other configuration parameters. The bonding driver must be a loadable kernel module because this is currently the only way to pass parameters to the driver and to configure it.

Linux kernel modules are loaded with the *modprobe* command. This command searches for the module in ***/lib/modules/2.4.25-elinos-133*** and uses the file *modules.dep* that is located there for module dependency checking. Because there is a fixed location of modules in Linux, the path to the module must not be given to *modprobe*, nor should the suffix of *.o* be given. Only the core driver name (in this case *bonding*) should be given to *modprobe*.

The object code for the bonding driver is located at:

```
/lib/modules/2.4.25-elinos-133/kernel/drivers/net/bonding/bonding.o
```

The first and most important parameter that the bonding driver needs is the **bonding policy**. This is controlled by the mode parameter. It is defined as follows:

Policy	Mode parameter value	Description
Round-robin policy	0	Transmit in a sequential order from the first available slave through the last. This mode provides load balancing and fault tolerance.
Active-backup policy (Default)	1	Only one slave in the bond is active. A different slave becomes active if and only if the active slave fails. The bond's MAC address is externally visible on only one port (network adapter) to avoid confusing the switch. This mode provides fault tolerance. **This is the default RAIDStor bonding policy.**
XOR policy	2	Transmit based on [(source MAC address XOR'd with destination MAC address) modula slave count]. This selects the same slave for each destination MAC address. This mode provides load balancing and fault tolerance.
Broadcast policy	3	Transmit everything on all slave interfaces. This mode provides fault tolerance.

Table 09: Bonding Driver Policy

Other parameters are defined as follows:

Parameter Name	Description	Values
miimon	Specifies the frequency in milliseconds that MII link monitoring will occur. A value of zero disables MII link monitoring. A value of 100 is a good starting point.	The default value is 0. **The RAIDStor defaults to 100.**

Parameter Name	Description	Values
downdelay	Specifies the delay time in milliseconds to disable a link after a link failure has been detected. This should be a multiple of *miimon* value, otherwise the value will be rounded.	The default value is 0. **The RAIDStor does not specify this value and thus uses the default.**
updelay	Specifies the delay time in milliseconds to enable a link after a link up status has been detected. This should be a multiple of *miimon* value, otherwise the value will be rounded.	The default value is 0. **The RAIDStor does not specify this value and thus uses the default.**
arp_interval	Specifies the ARP monitoring frequency in milliseconds. If ARP monitoring is used in a *load-balancing* mode (mode 0 or 2), the switch should be configured in a mode that evenly distributes packets across all links - such as *round-robin*. If the switch is configured to distribute the packets in an XOR fashion, all replies from the ARP targets will be received on the same link which could cause the other team members to fail. **ARP monitoring should not be used in conjunction with *miimon*.** A value of 0 disables ARP monitoring.	The default value is 0. **The RAIDStor does not specify this value, because it uses *miimon* instead.**
arp_ip_target	Specifies the IP addresses to use when *arp_interval* is > 0. These are the targets of the ARP request sent to determine the health of the link to the targets. Specify these values in ddd.ddd.ddd.ddd format. Multiple IP addresses must be separated by a comma. At least one IP address needs to be given for ARP monitoring to work. The maximum number of targets that can be specified is set at 16.	**The RAIDStor does not specify this value, because it does not use *arp_interval.***
primary	A string (***eth0, eth1***) to equate to a primary device. If this value is entered, and the device is on-line, it will be used first as the output media. Only when this device is off-line, will alternate devices be used. Otherwise, once a failover is detected and a new default output is chosen, it will remain the output media until it too fails. This is useful when one slave was preferred over another, i.e. when one slave is 1000Mbps and another is 100Mbps. If the 1000Mbps slave fails and is later restored, it may be preferred the faster slave gracefully become the active slave - without deliberately failing the 100Mbps slave. **Specifying a *primary* is only valid in *active-backup* mode.**	**The RAIDStor defaults to *eth0* as the *primary*.**
multicast	Integer value for the mode of operation for multicast support.	Possible values are: 0) Disabled (no multicast support). 1) Enabled on active slave only, useful in *active-backup* mode. 2) Enabled on all slaves, this is the default. **The RAIDStor does not specify this value and thus uses the default.**

Table 10: Bonding Driver Options

3.7.1.2.2 Initializing the Bonding Driver Network Interface

Initializing the bonding driver network interface is performed with the command ***ifconfig***. This command is used to set the IP address and to enable/disable it. The IP address is set using the *myraidstor* alias defined in ***/etc/hosts.*** The bonding network device is named ***bond0***. If a *netmask* or *broadcast* address is specified in ***/etc/hosts,*** then those will also be configured for this interface.

3.7.1.2.3 Enslaving the Ethernet Devices to the Bonding Driver

Enslaving the Ethernet devices to the bonding driver is performed with the command ***ifenslave***. This command is used to bind the Ethernet devices to the bonding driver. The Ethernet devices ***eth0*** and ***eth1*** are enslaved under the ***bond0*** device. The result of this step is that all Ethernet devices (***eth0, eth1***, and ***bond0)*** are all set to the same IP address and MAC address. There are no other options needed for this step.

3.7.1.2.4 Setting the Default Route

Setting the default route is performed with the command ***route.*** This command is used to make the default route point to the gateway. The IP address of the gateway is set using the *gateway* alias defined in ***/etc/hosts.*** This route was previously automatically set for the primary Ethernet device (***eth0***), but not for the new ***bond0*** device and thus it must be configured for this new device also. There are no other options needed for this step.

3.7.2 File System Exports with NFS and SMB/CIFS

These services are used to make public (export) the external partition(s). NFS is typically used by Linux/UNIX operating systems and SMB/CIFS is typically used by Microsoft Windows operating systems. These services are configured with ***/etc/exports*** and ***/etc/samba/smb.conf***, respectively. Additionally, a mount point file for each external partition must be created in the file system and an entry made in ***/etc/fstab*** to tell the operating system how to mount the external partitions.

SMB and NFS can share the same external partition as long as certain settings are present in the SMB setup file, ***/etc/samba/smb.conf*** in the [global] section. The required settings are as follows:

```
oplocks = False
level 2 oplocks = False
kernel oplocks = no
```

This disables SMB internal disk locking and data caching and enables kernel level disk locking that is also used by NFS.

3.7.2.1 NFS Export Options

Once the external partition has been locally mounted in the local file system, it needs to be exported or made public. To do this with NFS, each exported directory is added to the file ***/etc/exports***. When the NFS server starts, it reads this file to determine which directories to make public, which external clients are allowed to access this directory, and how they are to be exported. There can be multiple entries for the same exported directory that specify different clients and options. These entries can be on different lines or the same line. For example, one client could be allowed to mount the NFS export as read/write, whereas another client may only be allowed to mount it read only.

The format of ***/etc/exports*** is as follows:

```
directory_to_export clients(options) clients(options) ...
...
```

The *clients* are specified as follows:

Type of Client	Description	Examples
single host	This is the most common format. You may specify a host either by an abbreviated name recognized be the resolver, the fully qualified domain name, or an IP address.	buffy buffy.acttechnico.com 192.168.0.13

Type of Client	Description	Examples
netgroups	NIS netgroups may be given as @group. Only the host part of each netgroup members is considered in checking for membership. Empty host parts or those containing a single dash (-) are ignored. **Note: the RAIDStor does not currently run NIS, so this format is not supported at this time.**	@trusted
wildcards	Machine names may contain the wildcard characters * and ?. This can be used to make the exports file more compact; for instance, *.cs.foo.edu matches all hosts in the domain cs.foo.edu. However, these wildcard characters do not match the dots in a domain name, so the above pattern does not include hosts such as a.b.cs.foo.edu.	buf*.acttechnico.com *.acttechnico.com * The RAIDStor defaults to * to allow all clients to access the DRBD user disk.
IP networks	You can also export directories to all hosts on an IP (sub-)network simultaneously. This is done by specifying an IP address and netmask pair as address/netmask where the netmask can be specified in dotted-decimal format, or as a contiguous mask length (for example, either `/255.255.252.0' or `/22' appended to the network base address result in identical subnetworks with 10 bits of host). Wildcard characters generally do not work on IP addresses, though they may work by accident when reverse DNS lookups fail.	192.168.0.0/255.255.255.0 192.168.0.0/24

Table 11: NFS Clients

The *options* are a comma-separated list. The general export options are specified as follows:

Option	Description
secure, insecure	*secure* requires that requests originate on an internet port less than IPPORT_RESERVED (1024). This option is on by default. To turn it off, specify *insecure*. **The RAIDStor uses the default.**
rw, ro	Allow both read and write requests on this NFS volume. The default is to disallow any request which changes the file system. This can also be made explicit by using the *ro* option. **The RAIDStor defaults to *rw*.**
async, sync	*async* allows the NFS server to violate the NFS protocol and reply to requests before any changes made by that request been committed to stable storage (e.g. disc drive). **Using this option usually improves performance, but at the cost that an unclean server restart (i.e. a crash) can cause data to lost or corrupted.** If the highest fault tolerance is desired use *sync*. **The RAIDStor defaults to *sync*.**
no_wdelay, wdelay	*no_wdelay* has no effect if *async* is also set. The NFS server normally delays committing a write request to disc slightly if it suspects that another related write request may be in progress or may arrive soon. This allows multiple write to be committed to disc with the one operation, which improves performance. If an NFS server received mainly small requests, this behavior could actually reduce performance, so *no_wdelay* is available to turn it off. The default can be explicitly requested with the *wdelay* option. If the highest fault tolerance is desired use *no_wdelay*. **The RAIDStor defaults to *no_wdelay*.**

Table 12: NFS Options

3.7.2.2 NFS User ID Mapping

The NFS server bases file access control on the server machine on the *uid* and *gid* provided in each NFS RPC request. The normal behavior a user would expect is that they can access their files on the server just as they would on a normal file system. This requires that the same *uids* and *gids* are used on the client and the server machine. This is not always true, nor is it always desirable. Very often, it is not desirable that the *root user* on a client machine is also treated as *root* when accessing files on the NFS server. To this end, *uid 0* (root) is normally mapped to a different id: the so-called *anonymous* or *nobody uid*. This mode of operation (called `root squashing') is the default, and can be turned off with *no_root_squash*.

Option	Description
root_squash	Map requests from *uid/gid 0* to the anonymous *uid/gid*. Note that this does not apply to any other *uids* that might be equally sensitive, such as user *bin*.
no_root_squash	Turn off root squashing. This option is mainly useful for diskless clients. **The RAIDStor defaults to *no_root_squash*.**
all_squash	Map all *uids* and *gids* to the anonymous user. Useful for NFS exported public FTP directories, news spool directories, etc. The opposite option is *no_all_squash*, which is the default setting.
anonuid, anongid	These options explicitly set the *uid* and *gid* of the anonymous account. This option is primarily useful for PC/NFS clients, where you might want all requests appear to be from one user. **The RAIDStor sets both of these -2.**

Table 13: NFS User ID Mapping

3.7.2.3 Default NFS Options

The contents of the default RAIDStor ***/etc/exports file*** is as follows:

```
/raidstor   *(rw,sync,no_wdelay,no_root_squash)
```

This shows that the directory ***/raidstor*** is exported to any client (*). It allows reading and writing (*rw*); has no write delay (*no_wdelay*); and does not change a root client to another user (*no_root_squash*).

3.7.2.4 SMB/CIFS Export Options

Exporting external partitions using the SMB/CIFS protocol is controlled by the file ***/etc/samba/smb.conf***. When the SMB/CIFS server starts (called Samba), it reads this file to determine which directories to make public and how they are to exported. The Samba server is much more robust than NFS and has numerous options and configurations.

Rather than recreate the excellent documentation that already exists, please refer to:

http://www.samba.org/

for overall Samba server documentation.

3.7.2.5 Typical RAIDStor smb.conf File

On the RAIDStor, Samba is setup to allow any user to read and write the ***/raidstor*** partition. All files that are created are set to the *nobody* user and group.

Here is the default ***smb.conf*** file from the RAIDStor

```
[global]
    workgroup = Workgroup
    security = share
    netbios name = daytona
    log file = /var/log/samba/log.%m
    server string = ACT Technico
    create mode = 644
    directory mode = 755
```

```
    socket options = SO_RCVBUF=8192 SO_SNDBUF=8192
    oplocks = False
    level 2 oplocks = False
    kernel oplocks = no

[raidstor]
    path = /raidstor
    read only = no
    guest ok = yes
    force user = nobody
    force group = nobody
```

Here is the sample ***smb.conf*** file that forces users to login first to access the server

```
[global]
    workgroup = Workgroup
    security = user
    netbios name = raidstor
    log file = /var/log/samba/log.%m
    server string = ACT Technico
    create mode = 644
    directory mode = 755
    socket options = SO_RCVBUF=8192 SO_SNDBUF=8192
    oplocks = False
    level 2 oplocks = False
    kernel oplocks = no

[raidstor]
    path = /raidstor
    read only = no
    guest ok = no
    force user = nobody
    force group = nobody
```

Additionally, an SMB user must be added by executing the command:

```
smbpasswd –a user –p password
```

3.7.2.6 Restarting Samba

Once the smb.conf file has been updated restart samba from the console with the command “samba restart”. From a Windows Host on the same network search “My Network Places” for the netbios name specified above. Map the network drive to a drive letter and the RAIDStor path will be available at that volume.

3.8 High Availability Services

These services are used only on a dual blade RAIDStor to provide blade monitoring and disk mirroring across blades.

3.8.1 Distributed Replicated Block Device (DRBD)

The **D**istributed **R**eplicated **B**lock **D**evice (DRBD) server is a kernel module and associated scripts that provide a block device, which is designed to build high availability clusters. This is done by mirroring a whole block device via a dedicated network. Externally, it appears as a network RAID. In the RAIDStor, DRBD combines *RAID disk devices* across two RAIDStor blades making them act as a single disk device with RAID-0 (mirroring) capabilities to external clients. The result of this service is the installation of *DRBD disk device* files that are used by the higher layer services. Also an internal *DRBD disk name* is created which is used by the DRBD server to reference the *DRBD disk device.*

3.8.1.1 Quick Summary of How DRBD Works

Whenever a higher-level application, typically a journaling file system, issues an IO request, the kernel dispatches this request based on the target device. With DRBD registered for this request, it passes READ requests down the stack to the lower level device locally. WRITE requests are passed down the stack, too, but additionally they are sent over to the partner node.

Every time something changes on the local disk, the same changes are done at the same offset on the partner node's device. If a WRITE request is finished locally, a ``write barrier'' is sent over to the partner, to make sure that it is finished before another request comes in. Since later WRITE requests might depend on successful completion of previous ones, this is needed to assure strict write ordering on both nodes.

For more details see: http://www.drbd.org/fileadmin/drbd/publications/drbd-activity-logging_v6.pdf and http://www.drbd.org/fileadmin/drbd/publications/drbd_lk9.pdf

3.8.1.2 DRBD Configuration

DRBD is configured by ***/etc/drbd.conf.*** Another configuration file that comes into play with DRBD is ***/etc/hosts***; that will be discussed later. ***drbd.conf*** must be identical on each participating RAIDStor node.

drbd.conf is divided into several sections: a single *global* section and one or more *resource* sections. The *global* section has parameters that apply to all drbd resources. The *resource* sections describe each *DRBD disk device.*

There is one resource section for each *DRBD disk device*. Each *resource* section is named. This name is the *DRBD disk name* that is used internally by DRBD to reference the *DRBD disk device*.

Some parameters require an argument; others do not. All parameters that require an argument use only whitespace to separate the parameter from the argument.

Some parameters group multiple parameters together by brackets {}. This type of parameter is called a *directive*. They are *startup, disk, net, sync,* and *on*.

Some directives require a name argument. Use only whitespace to separate the directive and the name. They are *resource* and *on*.

The hash mark (#) is used to delineate comments from configuration settings.

A *resource* section typically consists of a few resource specific parameters, followed by *startup, disk, net,* and *syncer* directives, followed by DRBD node directives.

For complete details see: http://www.drbd.org/fileadmin/drbd/doc/0.7.0/en/drbd.conf.pdf

/etc/drbd.conf contains the IP address alias' of the primary and backup RAIDStor nodes that are participating in the array. These are defined in ***/etc/hosts.*** Each RAIDStor blade in the array must have a unique *IP Address Alias*; however other RAIDStor arrays may reuse the *IP Address Alias*, but must redefine the IP if they are located on a network with more than one RAIDStor.

Changing the host name of a RAIDStor blade is not recommended, because there are many script dependencies (used for SNMP) that use the currently defined hostnames.

To change the IP Address, the file ***/etc/hosts*** needs to be updated with the new *IP Address* on both nodes. The Heartbeat (covered later) and DRBD services must be restarted on both nodes or the system rebooted for new IP address to take effect.

3.8.1.3 Controlling DRBD

There are several commands available to control DRBD. Typically the common controls needed are to stop, start, unload, and load DRBD. All of these processes are automated on the RAIDStor and do not normally require manual intervention unless changing parameters. Also there is interdependency with Heartbeat, NFS, and SMB/CIFS, because these services need to be stopped before the DRBD driver is stopped. The order of shutdown is important, because you cannot stop the DRBD driver until you stop all *DRBD disk devices* and you cannot stop all *DRBD disk devices* until you first stop NFS and SMB/CIFS. Fortunately, it is easy to control this process with only two commands on each node

3.8.1.3.1 Stopping, Starting, Loading, and Unloading DRBD

To stop the all *DRBD disk devices*, NFS, and SMB/CIFS in the correct order, just stop Heartbeat on both nodes and wait until both nodes have finished. To do this, execute the following command on both nodes starting with the backup node first:

```
/etc/init.d/heartbeat stop
```

Wait until this command has finished on both nodes. Next, stop the DRBD driver and unload the DRBD module with this command on both nodes:

```
/etc/init.d/drbd stop
```

Now DRBD parameters can be modified.

To restart DRBD, execute the following command on both nodes:

```
/etc/init.d/drbd start
```

Wait until this command has finished on both nodes. DRBD will reparse ***drbd.conf*** at this point; watch for any errors, correct and restart if necessary. If successful, then:

```
/etc/init.d/heartbeat start
```

3.8.1.3.2 Forcing DRBD resynchronization

Another command for controlling DRBD is through the use of the ***drbdadm*** command. This command allows the user to set nearly any parameter on the fly. However, these changes are temporary and not persistent across DRBD restarts or reboots. It is primarily a testing tool. However there are two commands that could be used. ***drbdadm*** could be used to force a full resynchronization of the RAIDStor nodes. Typically, manual resynchronization is not needed, because DRBD performs this automatically when it deems necessary.

To execute a full resynchronization manually, execute the following command on the primary node:

```
drbdadm invalidate_remote all
```

Note, full resynchronization typically takes several hours to complete.

3.8.1.3.3 Booting a RAIDStor with DRBD Disabled

To boot a RAIDStor with DRBD and Heartbeat services disabled, the system needs to be booted at Linux runlevel 1. Normally the dual blade RAIDStor boots a Linux runlevel 4. When a RAIDStor is setup for the first time, it is set to start up at runlevel 1, to allow the operator to configure it first. Then it is restarted at runlevel 4 and all services are then started.

To change to runlevel 1, edit the file ***/etc/inittab*** as follows. Change the line which looks like this (typically line 4):

```
id:4:initdefault:
```

The **4** represents the runlevel. Change it to a 1 and reboot the RAIDStor.

3.8.1.3.4 Layout of the DRBD Disk

Currently, by default the dual blade RAIDStor is setup with two *DRBD disk devices:* a private device and a user exported device. The private device contains state information for the NFS and SMB/CIFS servers. The state information needs to be placed on a DRBD device so that it will survive a failover. In a normal Linux system, this state information is stored locally in ***/var/lib/nfs*** and ***/var/lib/samba*** and is different on every server. However, in a DRBD enabled system, the state information needs to be same for both RAIDStor blades. To manage this need, the *private* DRBD disk was created. This partition is not accessible to external clients, but is accessible to the active RAIDStor blade. The state file directories, ***/var/lib/nfs*** and ***/var/lib/samba*** are symbolic links to directories in the DRBD disk private area.

3.8.1.3.5 Mounting DRBD Disks in the Local File System

Once DRBD disks have been created, they need mount points in the local file system. For each RAIDStor blade, there will always be a mount point for every *DRBD disk device*. In a default RAIDStor, the mount points are:

```
/private
/raidstor
...
```

The mount point is filled with the contents of the DRBD disk only on the active RAIDStor blade. It will appear as an empty directory on the inactive RAIDStor blade, because only the active RAIDStor blade mounts the DRBD disk. It is important that the DRBD disk be mounted only on the active node. It should never be manually mounted on the inactive node or DRBD disk corruption could occur. Heartbeat is solely responsible for mounting and unmounting the DRBD disks on the RAIDStor nodes.

To tell Heartbeat how to correctly mount the DRBD disk, an entry is made ***/etc/ha.d/haresources*** for each DRBD disk. The format is as follows:

```
Filesystem::{DRBD_disk_device}::{mount_point}::{file_system_type}::{options}::{options}::...
...
```

Parameter	Device	Notes

DRBD_disk_device	`/dev/drbd/0`
	`/dev/drbd/1`
	`...`
	`/dev/drbd/N`	

mount_point	`/private`
	`/raidstor`
	`...`	

type_of_file_system	ext3	Future RAIDStor's may support other journaling file systems, but currently only ext3 is supported.
options Note: individual options are double colon separated with no white space between them.	noauto	Do not automatically mount the file system on bootup. Heartbeat handles mounting and unmounting. **Do not use this option.**
	rw, ro	Mount the file system read/write or readonly. Typically the user will want it mounted read/write. **RAIDStor defaults to read/write.**
	sync, async	Disk I/O is done synchronously or asynchronously. **The RAIDStor defaults to *sync* for fault tolerance**.
	exec, noexec	Permit execution of binaries or not. **The RAIDStor defaults to *exec*.**
	dev, nodev	Interprete char and block devices as special or not. The **RAIDStor defaults to *nodev*.**
	atime, noatime	Update inode access time for each access or not. The file system is faster with *noatime*. **The RAIDStor defaults to *atime*.**
	…	There are other options, but they don't appear to be important.

Table 14: DRDB Mount Options

3.8.2 Heartbeat

The Heartbeat server is used to monitor the RAIDStor blades and inform the cluster when one of them dies and to start/stop services on the blades. It includes the ***fake*** server for IP address takeover. The RAIDstor uses Heartbeat to create a common public IP address that is shared by the two blades and to start/stop DRBD, NFS, SMB/CIFS and any other service that is to run on the active

blade. It is configured with ***/etc/ha.d/ha.cf***, ***/etc/ha.d/haresources,*** and ***/etc/ha.d/authkeys.*** Other configuration files that come into play with Heartbeat are ***/etc/init.d/bonding.sh***, ***/etc/hosts***, and ***/etc/drbd.conf***.

The RAIDStor uses redundant Ethernet interfaces to communicate between nodes. This provides for loss of a single network path or single switch failure. Optionally the serial ports could be connected with a null modem cable and Heartbeat could use that as a third backup communication path in case both networks fail. However, that is probably not necessary, because it both networks fail, then neither RAIDStor node would be accessible from outside clients and because the networks have failed, the dual resource problem would not be an issue.

3.8.2.1 Heartbeat Configuration

There are three files needed to configure Heartbeat. The first is ***/etc/ha.d/ha.cf***. This file tells Heartbeat what types of media paths to use and how to configure them. There are a lot of options in this file and the order of some of the options does matter in some cases. The hash mark is used to mark a comment. The default setting is used for any option that is not specified.

Although not required, ***/etc/ha.d/ha.cf*** should be that same on both RAIDStor nodes. There is really no reason to set different options on each node and to support that would be complex and invite trouble.

3.8.2.2 Heartbeat Resources

The Heartbeat resource file ***/etc/ha.d/haresources*** is used to setup the common IP addresses, declare primary nodes, and indicate services to run on those nodes. This file must be the same on both RAIDStor nodes.

It is entirely possible to have different nodes be declared as the *primary* node for different services. The only reason to do that would be for performance reasons such as load sharing across nodes. However, this is not a good idea for the dual node RAIDStor where high availability is the goal. So the default setup on the RAIDStor is to have one node declared as *primary* for all services and the other as *backup*.

The format of ***/etc/ha.d/haresources*** is as follows:

```
primary_host_name scriptname::argument ...
```

All scripts must be located in ***/etc/ha.d/resource.d, /etc/ha.d/rc.d,*** or ***/etc/init.d***. Scripts are passed arguments by using a "::" to separate each argument. All scripts must support *start, stop,* and *status* as the first argument. Scripts located in ***/etc/ha.d*** are Heartbeat related scripts and those located in ***/etc/init.d*** are system service scripts.

3.8.2.2.1 IPaddr Script

The common IP address is set with the Heartbeat related ***IPaddr*** script. The only argument is the IP address. An IP address alias may NOT be used. The IP address in dotted decimal notation must be given. A netmask is attached with a "/" and a gateway may be attached to the netmask with another "/". Typically the netmask and gateway are not needed. Because the ***IPaddr*** script is so common, the script name does not need to be specified, just the IP address. The following are equivalent:

```
primary IPaddr::192.168.0.200
primary 192.168.0.200
```

3.8.2.2.2 DRBD Script

DRBD is started with the Heartbeat related ***drdbdisk*** script. The only argument is the *DRBD disk name* specified in ***/etc/drbd.conf***. Adding that to ***haresources*** we now have:

```
primary IPaddr::192.168.0.200 \
   drdbdisk::private \
   drbddisk:raidstor
```

3.8.2.2.3 Filesystem Script

DRBD file system is mounted with the ***Filesystem*** script. The arguments are *device driver, mount point, file system type,* and *mount options.* Adding that to ***haresources*** we now have:

```
primary IPaddr::192.168.0.200 \
```

```
drdbdisk::private  Filesystem::/dev/drbd/0::/private::ext3 \
drdbdisk::raidstor Filesystem::/dev/drbd/1::/raidstor::ext3
```

3.8.2.2.4 NFS and SMB/CIFS Server Scripts

NFS server is started with the system ***nfs-server*** script. There are no arguments. SMB/CIFS is started with the system ***samba*** script. Adding those to ***haresources*** we now have:

```
primary IPaddr::192.168.0.200 \
  drdbdisk::private  Filesystem::/dev/drbd/0::/private::ext3 \
  drdbdisk::raidstor Filesystem::/dev/drbd/1::/raidstor::ext3 \
  nfs-server samba
```

3.8.2.2.5 led_nas Script

This script is used to turn on/off the *NAS* LED on the front panel. It has no arguments. Adding that to ***haresources*** we now have:

```
primary IPaddr::192.168.0.200 \
  drdbdisk::private  Filesystem::/dev/drbd/0::/private::ext3 \
  drdbdisk::raidstor Filesystem::/dev/drbd/1::/raidstor::ext3 \
  nfs-server samba led_nas
```

3.8.2.3 Heartbeat Authentication Keys

Heartbeat authentication is specified by the file ***/etc/ha.d/authkeys***. This file must be the same on both RAIDStor nodes. There are three types of authentication methods available: *crc*, *md5*, and *sha1*. If Heartbeat runs over a secure network, such as the crossover cable, use *crc*. This is the cheapest method from a resources perspective. If the network is insecure, but want to minimize CPU resources, use *md5*. Finally, the best authentication without regard for CPU resources is *sha1*. It's the hardest to crack. **The RAIDStor defaults to *crc*.**

The format of the file is as follows:

```
auth number
number authmethod authkey
```

For *sha1*, a sample ***/etc/ha.d/authkeys*** could be:

```
auth 1
1 sha1 key-for-sha1-any-text-you-want
```

For *md5*, you could use the same as the above, but replace "*sha1*" with "*md5*".

Finally, for *crc*, use:

```
auth 2
2 crc
```

Whatever index you put after the keyword **auth** must be found below in the keys listed in the file. If you put "auth 4", then there must be a "4 signaturetype" line in the list below.

To make sure its permissions are safe, use a permission octet of 600.

3.9 USB

The RAIDStor includes 3 USB ports. One is located on the front panel and 2 are on the rear transition module. All USB ports are Version 1.1. The USB interface is setup to support storage media including flash drives, hard drives, and CDROM drives. Since the RAIDStor does not include a graphical user interface (GUI), USB keyboards and mice are not supported. The USB interface is setup to be Plug and Play. When a device it is automatically detected and a device driver is created. A message similar to this will be displayed on the serial console:

```
hub.c: new USB device 01:0e.0-1, assigned address 2
```

```
Manufacturer: SanDisk Corporation
Product: Cruzer Mini
SerialNumber: 000012986
scsi4 : SCSI emulation for USB Mass Storage devices
  Vendor: Generic   Model: STORAGE DEVICE    Rev: 1.22
  Type:   Direct-Access                      ANSI SCSI revision: 02
Attached scsi removable disk sde at scsi4, channel 0, id 0, lun 0
sde: Write Protect is off
 /dev/scsi/host4/bus0/target0/lun0:<7>usb-storage: queuecommand() called p1
```

This device can then be mounted and accessed like this:

```
[90 root@primary ~] mount /dev/scsi/host4/bus0/target0/lun0/part1 /mnt
[91 root@primary ~] ls /mnt
bootex.log    burnin~1.txt  deskto~2    finder.dat  pro2kxp.exe  wardsb~1
burnin~1.exe  deskto~1      fileid.dat  oceanc~1    virexs~1
```

When done with the device, umount it and remove it.

```
[92 root@primary ~] umount /mnt
```

Once removed, you will see this message on the serial console:

```
usb.c: USB disconnect on device 01:0e.0-1 address 2
```

3.10 Firmware

The RAIDStor ships with an on-board firmware monitor that can be used to configure, test and boot the board. As shipped from the ACT/Technico factory the RAIDStor is configured to auto-boot upon power up or on a board reset. To enter the firmware monitor connect the serial cable supplied by ACT/Technico to a standard DB9 serial port on a PC or a Linux terminal and press the <**escape**> key quickly after power up. This will interrupt the boot sequence. A screen similar to the following will be displayed.

```
 _____________________ Secondary MenMon for the D303 Version 0.95 _____________
|                                                                               |
|             Parts of this code based on Motorola's Dink32                     |
|                  Created  Oct 24 2005     10:36:54                           |
|_______________________________________________________________________________|
|    HW Revision: 01.02.00            |            CPU: MPC8245     (MAP B)     |
|  Serial Number: 35                  |  CPU/MEM Clock: 400 / 132.0 MHz         |
|    Board Model: D303-00             |    DIMM Module: 256 MB Setup: 333       |
|  CPLD Revision: 0x07                |                                         |
 \_____________________________________________________________________________/

Press 'ESC' to setup/MENMON
Selftest running...
                    === PCI ===
MPC8245             bus 0x0, dev 0x00       ==> OK
ALI1543 PCI2ISA     bus 0x1, dev 0x02       ==> OK
ALI1543 IDE         bus 0x1, dev 0x0B       ==> OK
ALI1543 PMU         bus 0x1, dev 0x0D       ==> OK
ALI1543 USB         bus 0x1, dev 0x0E       ==> OK
SiI3114 SATA        bus 0x0, dev 0x18       ==> OK
Enet 82541ER I      bus 0x0, dev 0x17       ==> OK
Enet 82541ER II     bus 0x0, dev 0x1A       ==> OK
                    === SMB ===
SO-DIMM SPD         ==> OK
RTC                 ==> OK
LM75                ==> OK
                    === FLASH ===
CHECKSUM            ==> OK

Shuttle 0: PRESENT
Shuttle 1: PRESENT

MenMon>
```

From the MenMon prompt the **help** command can be used to display the commands available to the user.

```
MenMon> help
```

Expanded help for the available commands can be obtained by typing the command you wish to run after the help command. For example, to learn what diagnostic commands are available type **help diag**.

```
MenMon> help diag
```

RAIDStor hardware/firmware configuration information is stored in a serial EPROM which can be read/written from the MemMon> prompt. To display the current information simply type **ee.**

```
MenMon> ee
```

The current configuration for this blade will be displayed on the serial console. The default configuration programmed at ACT/Technico is as follows.

```
MenMon> ee

Console Baudrate:                       9600
PS/2 keyboard layout:                   84
```

```
VESA video mode:                          0x0
Bootstrapper address:                     $0
Startup String:                           DBOOT
Selftest mode:                            EXTENDED
Selftest halt mode:                       NO
Selftest message mode:                    NO
IP address/subnet mask:
Gatway IP address:
Host IP address:
Machine NetName:
Bootfile:pImage
Kernel Parameters: root=/dev/md/0 rw
VxWorks boot line:                        (0,0):pImage
s='ip=192.168.0.201::192.168.0.1::raidstor_pri:eth0 root=/dev/md/0 rw'
NBOOT retries:                            0
CPU->PCI store gathering:                 enabled
PCI->CPU memory specul. reads:            enabled
```

Configuration options are changed via the ee-xxxx command, where xxxx is the option you wish to modify. To modify the Startup String you would type

MenMon> **ee-startup yourStartupString**
MenMon> **ee-startup dboot** **(default string, nboot for network boot)**

To clear an option use "-" as in

MenMon> **ee-startup** –

3.10.1 Key Boot Options

A user might wish to inspect and/or modify the following MenMon boot options.

ee-startup

The *dboot* setting tells the firmware monitor to boot from the compact flash. The *nboot* setting tells the firmware monitor to boot over the network using the tftp protocol. Consult with ACT/Technico engineering if you wish to store your kernel on the network.

ee-bootfile

The default configuration is to boot a file name *pImage* from the Compact Flash. Currently this is the only file stored on this device. *pImage* is a Linux kernel configured to boot on a PowerPC.

ee-kerpar

The *kerpar* string is the list of options passed to the Linux kernel on startup. For a complete list of options check the documentation directory in a Linux 2.4 kernel source tree. The RAIDStor's root directory, ***/dev/md/0*** is the Linux raid device.

The default value of *kerpar* is:

```
/dev/md/0 rw
```

To boot the backup copy of the root filesystem change ee-kerpar to /dev/md/1 rw from the firmware prompt:

MenMon> *ee-kerpar root=/dev/md/1 rw*

Optionally the IP address may be specified. Note however, this is only needed if loading the kernel from the network (ee-startup = *nboot*).

```
ip=192.168.0.201::192.168.0.1:255.255.255.0:primary:eth0 root=/dev/md/0 rw
```

To use bootp:

```
ip=bootp root=/dev/md/0 rw
```

Any network value specified in ee-kerpar, is only used to boot the Linux kernel (pImage). Once booted, however these network parameters are over written by the values in the ***/etc/hosts*** file.

These three options need to be set correctly to enable automatic booting on a reset or power cycle.

3.11 Testing the RAIDStor

The following initial testing is suggested to insure the RAIDStor is functional in your system.

Mount the NAS_IP: /raidstor directory from a Linux or Unix Machine via NFS and insure you can read/write files. On a dual blade RAIDStor, after a failover, other than a short 30 second delay, there should be no interruption of any kind, because NFS uses the connectionless UDP protocol. A read and/or write request will simply block until the data is available.

From a Windows machine you should be able to mount a directory \\Daytona\raidstor. This directory should appear in your network neighborhood. On a dual blade RAIDStor, after a failover, the connection will be interrupted because SMB/CIFS uses the connection based TCP protocol. To re-establish the connection, just remount the directory \\Daytona\raidstor as explained in Section 3.7.2.5.

3.11.1 Single Blade RAIDStor

3.11.1.1 Initial Setup

Install the RAIDStor blade and let it boot. When the NAS LED is illuminated the RAIDStor is ready.

3.11.1.2 Shuttle Hotswap Test

1) Remove a shuttle as outlined in Section 3.2.
2) Wait about 30 seconds for the shuttle to go offline. Test that you can access any files you have in the /raidstor directory via NFS or Windows.
3) Replace the shuttle as outlined in Section 3.2.
4) The shuttle will complete the resync and come back on-line. The time to fully rebuild a shuttle will vary with the size of the hard drive and the amount of network activity. It is possible to fully rebuild a shuttle with the network effectively saturated with user requests to the public partition.

3.11.2 Dual Blade RAIDStor

3.11.2.1 Initial Setup

Install the two RAIDStor blades and let them boot. When the HRTB (heartbeat) LED is blinking on both blades and the NAS LED is illuminated on a single blade the RAIDStor is ready.

3.11.2.2 Shuttle Hotswap Test

1) Remove a shuttle as outlined in Section 3.2.
2) On a RAID-0 blade in about 20 to 30 seconds the backup blade will assume the active role and will take control of the public NAS address if the blade is configured for RAID level 0. At this point you should again be able to access any files you have in the /raidstor directory via NFS or Windows. On blades configured for RAID1 or RAID5 removing a single drive will not cause a failover.
3) Replace the shuttle as outlined in Section 3.2.
4) In about 2 - 4 hours the Primary Blade will assume the active role and will take control of the public NAS address. Verify this with the NAS LED.

3.11.2.3 Blade Failover Test

1) Press the Ejector switch on the RAIDStor that has the NAS LED illuminated. Alternately you could use the Web interface to HALT the Active blade or Stop Heartbeat on the Active Blade. You could also just yank the primary blade out without warning to simulate a real world failure. On a soft shutdown the time for recovery is short. When the primary blade is brought back on-line only changes need to be sent from the backup. On a crash, the entire /raidstor partition must be re-synced. This is very time consuming.
2) After about 30 seconds the Blue HOT SWAP LED will illuminate.
3) In about 20 to 30 seconds the backup blade will assume the active role and will take control of the public NAS address. At this point you should again be able to access any files you have in the /raidstor directory via NFS or Windows.
4) After verifying that the Active blade will failover to the Backup you can simply reboot the blade you just shut down. The reset button on the top left corner of the RAIDStor can be used to cause a re-boot. Remember to raise the Ejector handle if you pushed it down.
5) Upon boot, the primary blade (which is now in the backup role) will contact the currently Active blade and rebuild (re-sync) the hard drives. This is indicated by the GP LED and the Green Drive LEDS, which will be illuminated.
6) When the re-sync is complete, the primary blade will again take control of the NAS IP addresses and responsibility for the exported file system

4 Control and Status Interfaces

This section covers all available control and status interfaces to the RAIDStor. A control and status interface is any method for accessing a RAIDStor to modify behavior or check current status. This section also discusses the level of accessibility and security for each interface.

4.1 Command Line Interface

A fully functional Linux command shell implements the command line interface of the RAIDStor. Available command shells are: ***bash*** and ***sh***, but typically it is ***bash***. See the ***bash*** man page for more information on usage. The actual command shell used by each user can be configured from the command line (see man page for ***passwd***).

Because the command line is not limited in any way, all configuration files and commands are accessible. However, to maintain configuration coherency on dual blade RAIDStors, users are encouraged to use the Web/SNMP Interface. There is currently no configuration synchronization between the command line interface, the Web/SNMP Interface, or blades in a dual blade RAIDStor when using the command line. Command line changes of system configuration files are only detected during startup of the SNMP daemon, startup of networking, or system boot. If configuration changes relevant to the Web/SNMP Interface must be made at the command line, the SNMP daemon should be stopped and started following the changes or both blades in a dual blade system should be rebooted.

Example command line procedures are given here Configuration Procedures using the Command Line; however, configuring a RAIDStor from the command line can be quite complex and is only recommended for experienced users. Please note that is it easy to make erroneous settings when using the command line interface that will render the system inoperable.

4.1.1 Serial Console Port

The serial console port is available immediately upon power up of the SBC. This is the only interface that provides access to the MENMON firmware command line and firmware status messages. This interface also displays OS boot messages during OS initialization and shuttle insertion/removal status messages. Once the system boots completely, this interface provides a typical login prompt (requiring a username and password). The console port is configured for 9600 baud. A console cable ships with each RAIDStor. Details of the console cable can be found here.

Note that each SBC in a dual blade system has a separate serial port terminal, so a cable must be connected to each SBC to monitor both blades. Any serial port terminal client can be used to connect to the serial port interface, including minicom on Linux and HyperTerm on Windows.

Serial Port Login Session

```
primary login: root
Password: ********
login[55]: root login  on `console'

[501 root@primary ~]
```

4.1.2 Telnet/Rlogin/SSH

These services are available following network interface initialization in Linux. Currently only ***telnet*** and ***ssh*** are supported (***rlogin*** is not. Telent and ssh are enabled by default on the RAIDStor.

4.2 Web Interface

The web interface is the preferred method for connecting to, configuring, and monitoring status of the RAIDStor. Each page has the following:

- The ACT/Technico Logo (that links to the ACT home page).
- A link to this manual is available on each page.
- Email link to request technical support from ACT/Technico. OEM customer may want to change this link to another value.
- Home and logout buttons.

- Whenever possible, the content of the pages is displayed on *one* screen, that is, no scroll bars.
- A side toolbar is provided to switch between system pages.

4.2.1 Home Page (Login)

This page prompts for the administrative password to login. Following a successful login, the Home Page is displayed as shown in Figure 13: Web Page – Home. Current defaults for username and password:

username: **admin**

password: **admin1423**

The entries on the left side of the page are links to the various web pages.

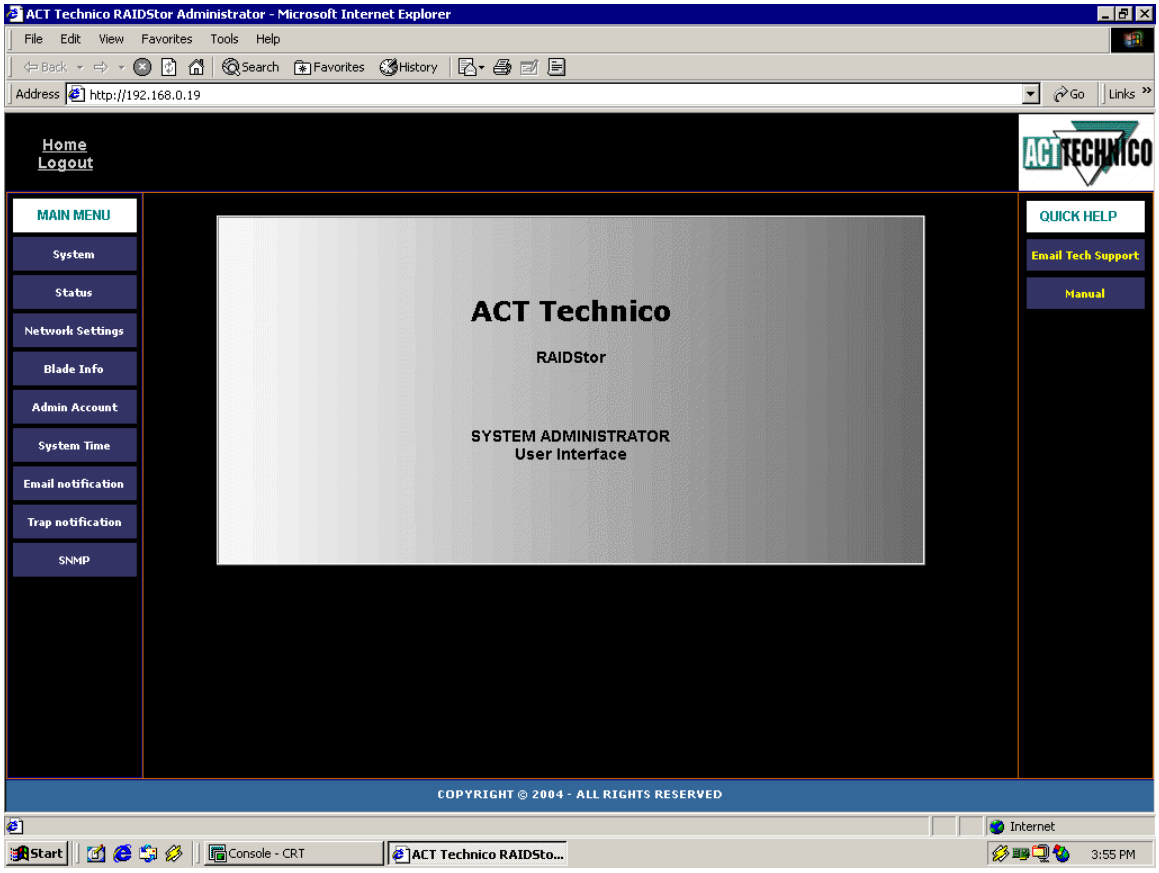


Figure 13: Web Page – Home

4.2.2 System Page

The System Page provides some basic information about the RAIDStor and provides the user with buttons to shutdown or reboot a RAIDStor. On a dual blade RAIDStor the buttons on this page shutdown or reboot both blades if accessed from the shared public IP address of the RAIDStor pair. The system health is displayed by lights.

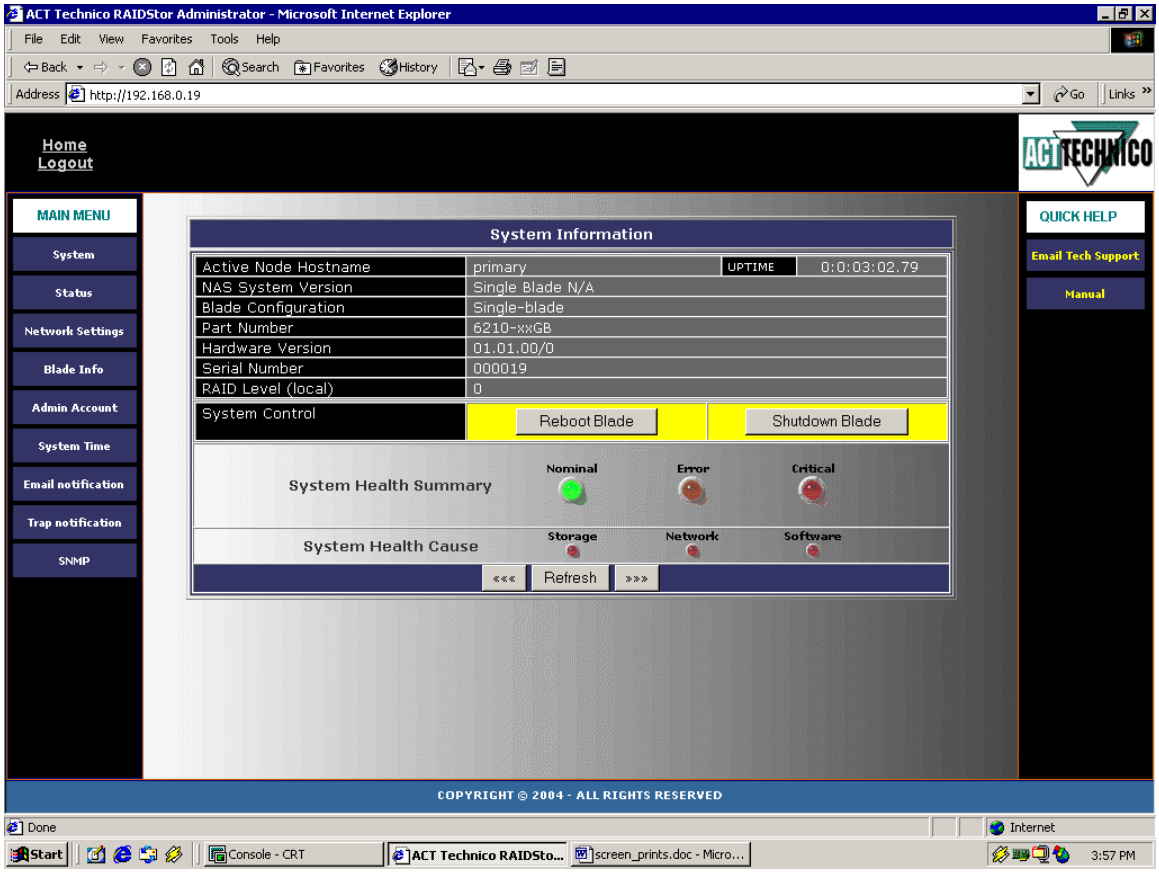


Figure 14: Web Page – System – Single Blade

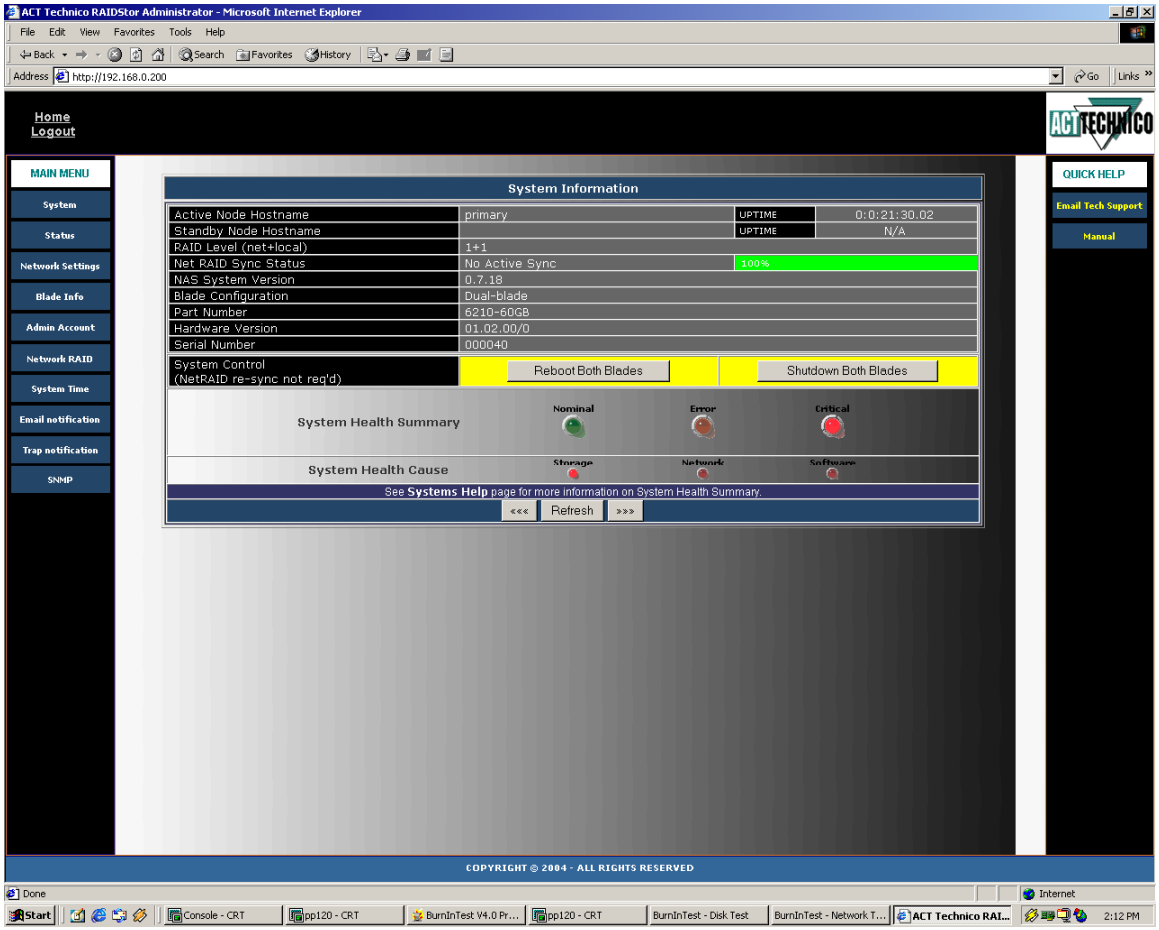


Figure 15: Web Page – System -- Dual Blade

4.2.2.1 System

This page provides basic system information, including:

- The NAS system version is the release of DRBD on this system. DRBD is not used on a single blade RAIDStor.
- Blade configuration: single or dual.
- System health summary. Information related to System Health will be provided on the Status page and the Blade Info page. This indicator is a blinking LED that can be green, yellow, or red with these meanings:
  - Nominal (green) - System is fully operational.
  - Error (yellow) - System is operating at a reduced level of redundancy.
  - Critical (red) - System is operating without redundancy.
- The Blade Serial Number is read from the EEPROM.
- Uptime is the amount of time the SNMP agent has been up. Stopping and starting the agent will reset this value.

4.2.2.2 System Health Summary

4.2.2.2.1 Single Blade

- On a single blade RAIDStor the green LED indicates that all disk shuttles are online and both Ethernet ports and the bonding driver are functional.
- If an Ethernet port or a disk shuttle is down the critical LED will be blinking and the associated health summary cause LED will be on to indicated that the system is running with a reduced level of redundancy.
- The software LED and the error LEDs are not used on a single blade webpage.

4.2.2.2.2 Dual Blade

- On a dual blade RAIDStor the green LED indicates that both blades are operation (meaning the DRBD and Heartbeat are both functional), all disk shuttles, both Ethernet ports and the bonding driver are functional on both blades.
- On a dual blade the software LED is used to indicate that DRDB and/or Heartbeat is stopped.
- On a dual blade RAIDStor the yellow error LED will be illuminated if a single shuttle on a RAID-1 or RAID-5 blade is down.
- If a disk shuttle on a RAID-0 blade is down then the critical LED will be illuminated as the blade is effectively down and no storage redundancy is available.
- If one of the blades are down the critical LED will be illuminated.

4.2.3 Status Page

The example below is from a single blade RAIDStor configured for RAID0. A Dual Blade example follows. The Status page displays status for the following items:

- Network interfaces (link up or down).
- Primary/Secondary blade status in dual blade system.
- The temperature of the RAIDStor is read from an on-board sensor.
- The status of the two Ethernet ports and the bonding driver is displayed.
- Heartbeat is not used on a single blade.
- The status of each disk drive is displayed.
- The public IP address on a single blade is the same as the address of the bonding port (bond0). On a dual blade this is actually the address of bond0:0.

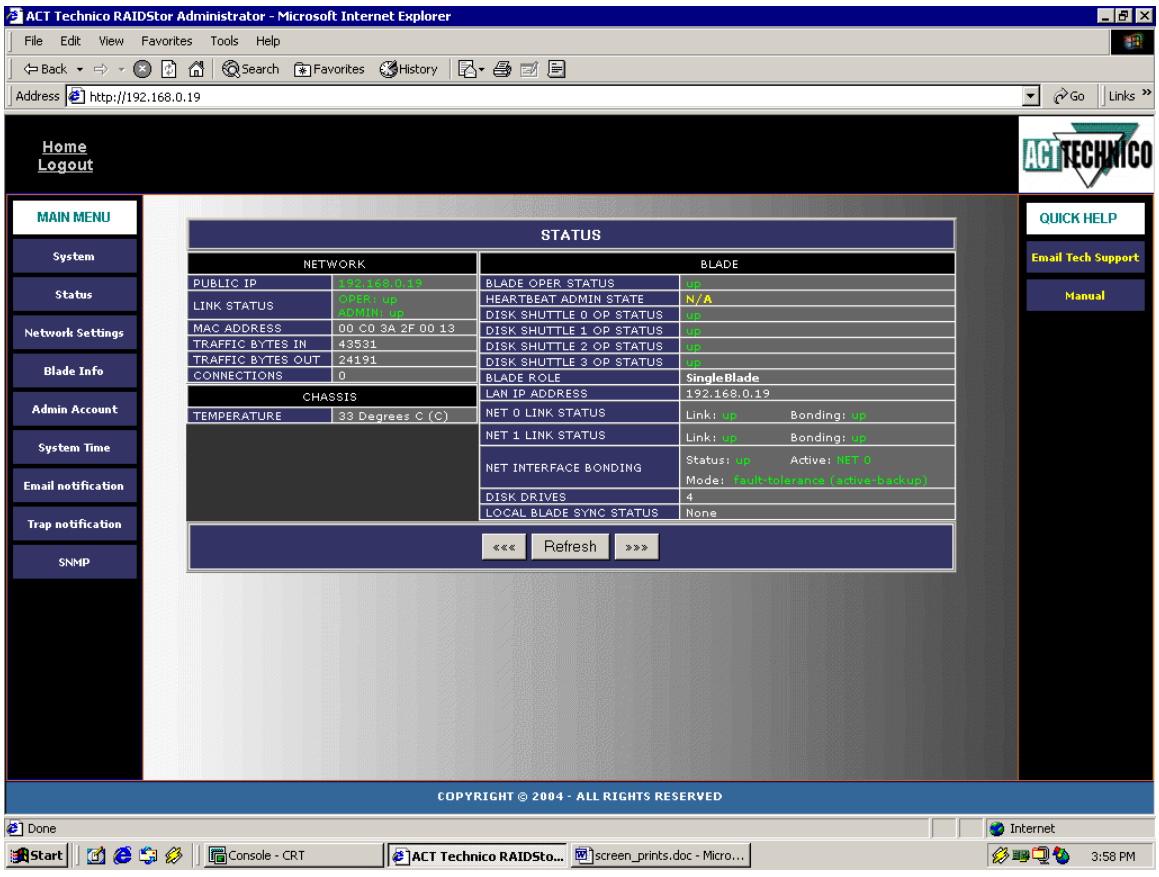


Figure 16: Web Page – Status – Single Blade

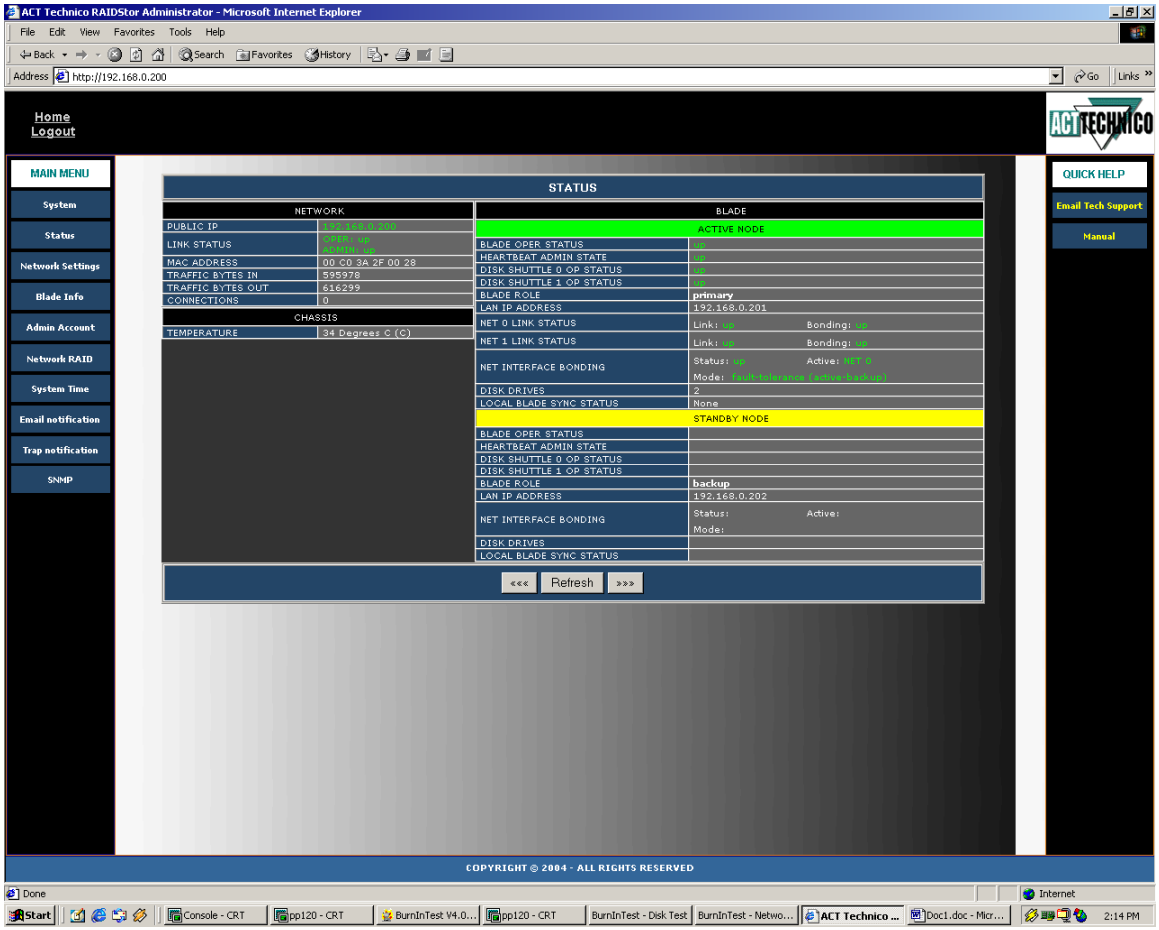


Figure 17: Web Page -- Status -- Dual Blade

4.2.4 Network Settings Page

This page provides current values for important network settings. It also provides the ability to modify some settings. On a dual blade it provides the ability to start and stop heartbeat. Examples of both Single and Dual Blade RAIDStor pages are shown below. Some of the parameters shown on this page are:

- The seven buttons on the top of the web page require a Macromedia flash plugin to function correctly. This plugin is available for free on the web. These buttons provide access to standard IP MIB values. The page can be used without the plugin if it is not available.
- Single IP address for single blade or the three IP addresses used in dual blade systems. On a single blade changing either IP address will change both.
- System Hostname, the hostname in the column with the Virtual IP address is a virtual name assigned to the public NAS.
- Domain name
- MAC address
- Default gateway
- DNS server(s)

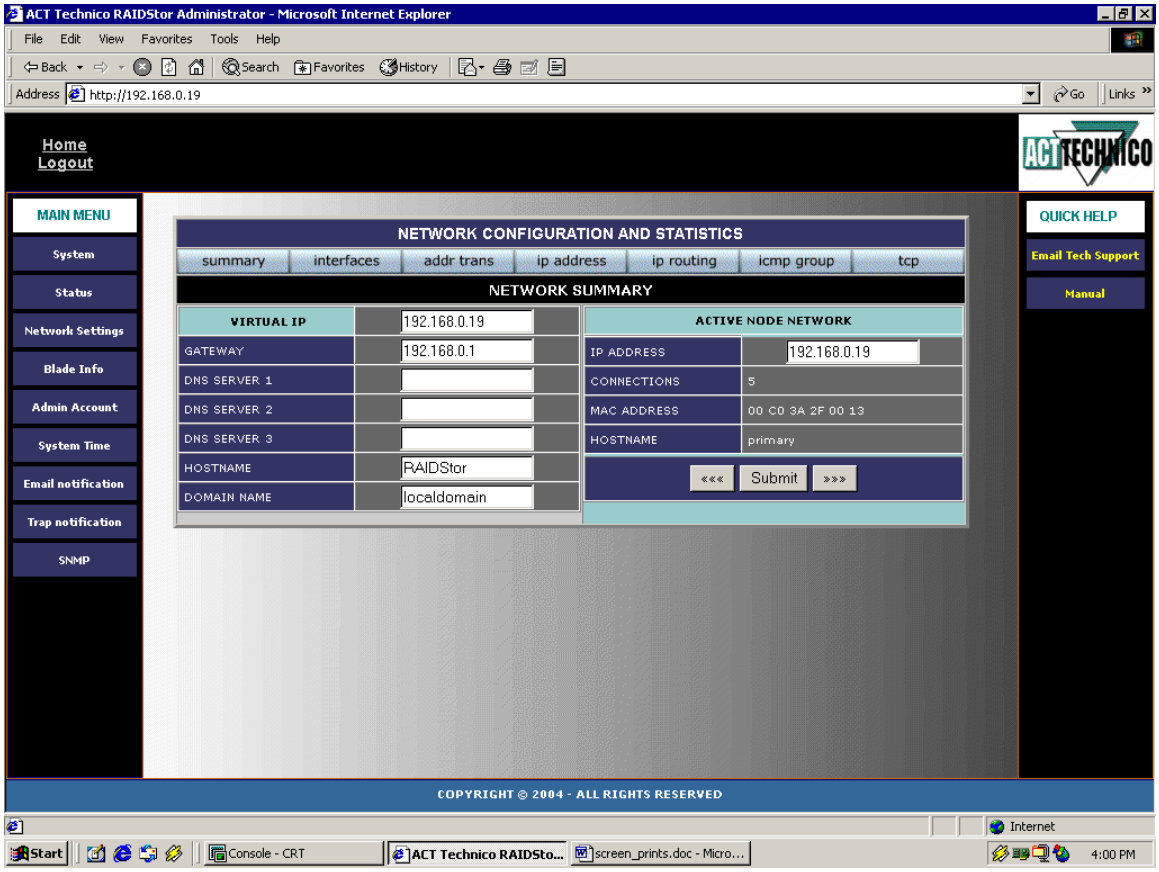


Figure 18: Web Page - Network Settings – Single Blade

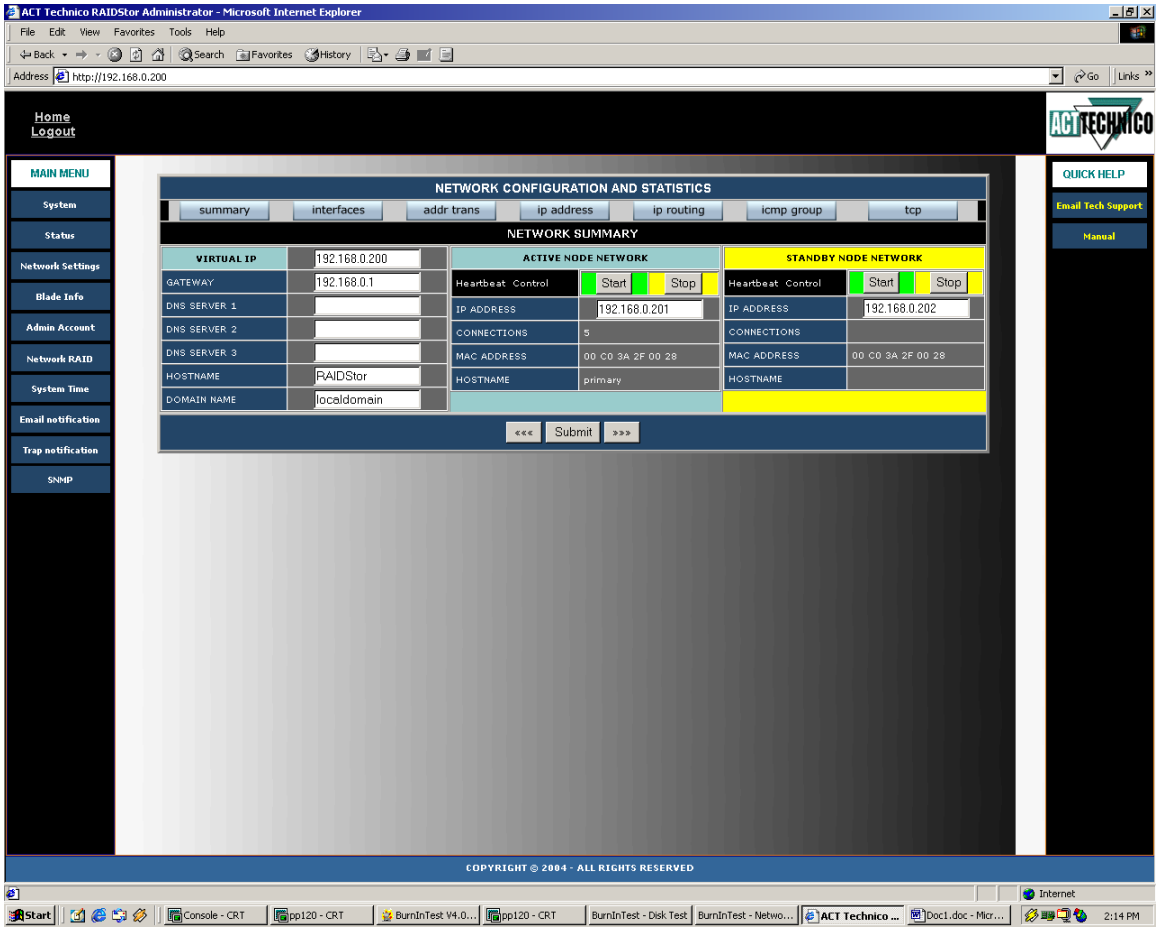


Figure 19: Web Page -- Network Settings -- Dual Blade

4.2.5 Blade Info Page

This page provides information on one or both blades in a dual blade system. Examples of both Single and Dual Blade pages are shown below. For a dual blade there is a selector in the upper left corner of the web page that is used to select the primary or secondary blade. Heartbeat can also be stopped and started on a dual blade. Some of the parameters shown on this page:

- The part number is read from /usr/share/snmp/raidstor.cfg, not the EEPROM, so it is not guaranteed to be accurate.
- The serial number and hardware version are read from the EEPROM. They can also be accessed from the /proc filesystem.
- Blade and Disk Shuttle Status are displayed via lights.
- The software version is read from /usr/share/snmp/raidstor.cfg. It consists of the date of the software release loaded on this blade and the "checkin" number from the revision control system used by ACT/Technico.
- The local blade sync percent is an estimate of the amount of the hard drive that has been synced. This is updated by the SNMP agent code about every 30 seconds.
- The free space available in the root partition and the /raidstor partition is displayed. If the name of the public partition is changed by the user, the PERL script called by the SNMP agent would need to be adjusted. These scripts are available in the /usr/local/bin directory on the RAIDStor. The names of the scripts in this directory generally match the names used by the SNMP agent (rsGetRsBladeFreeSpace*).

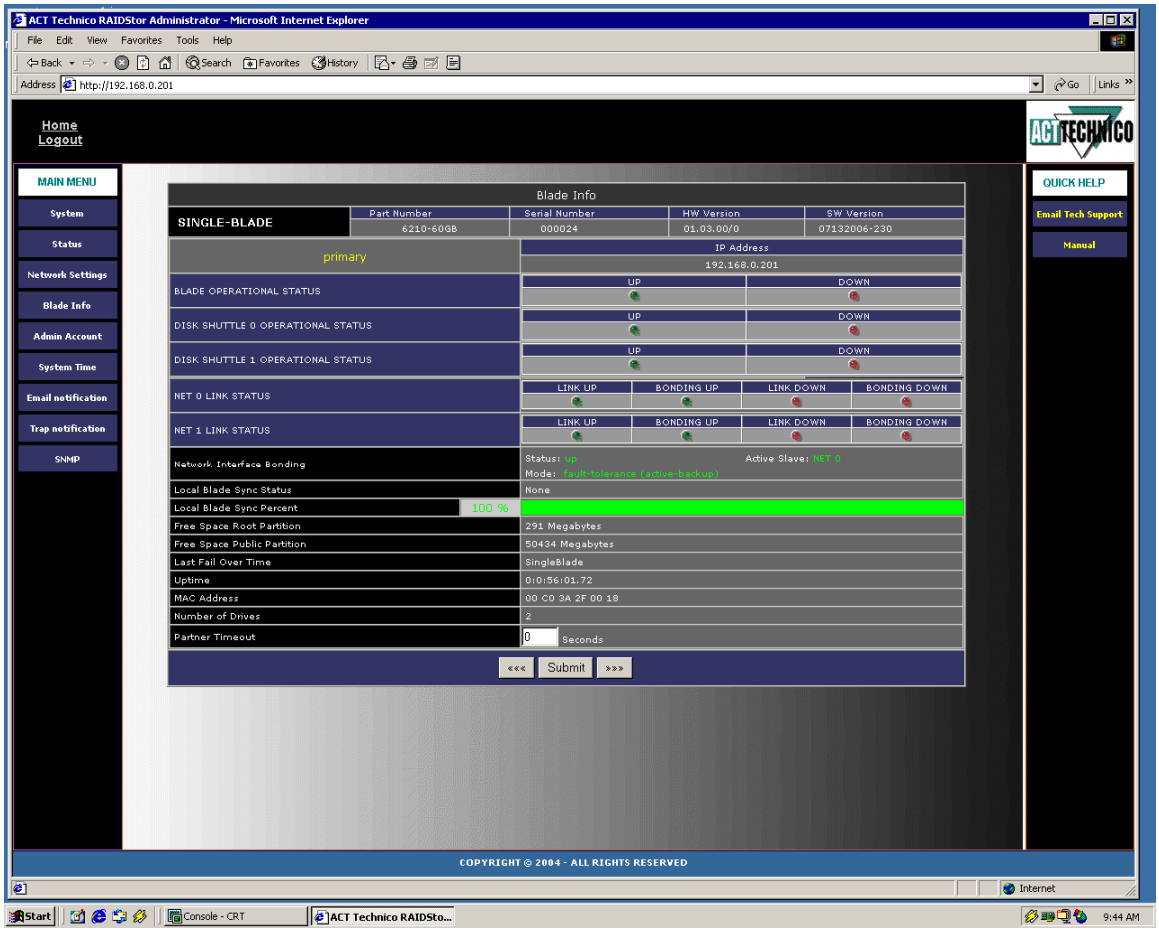


Figure 20: Web Page – Blade Info – Single Blade

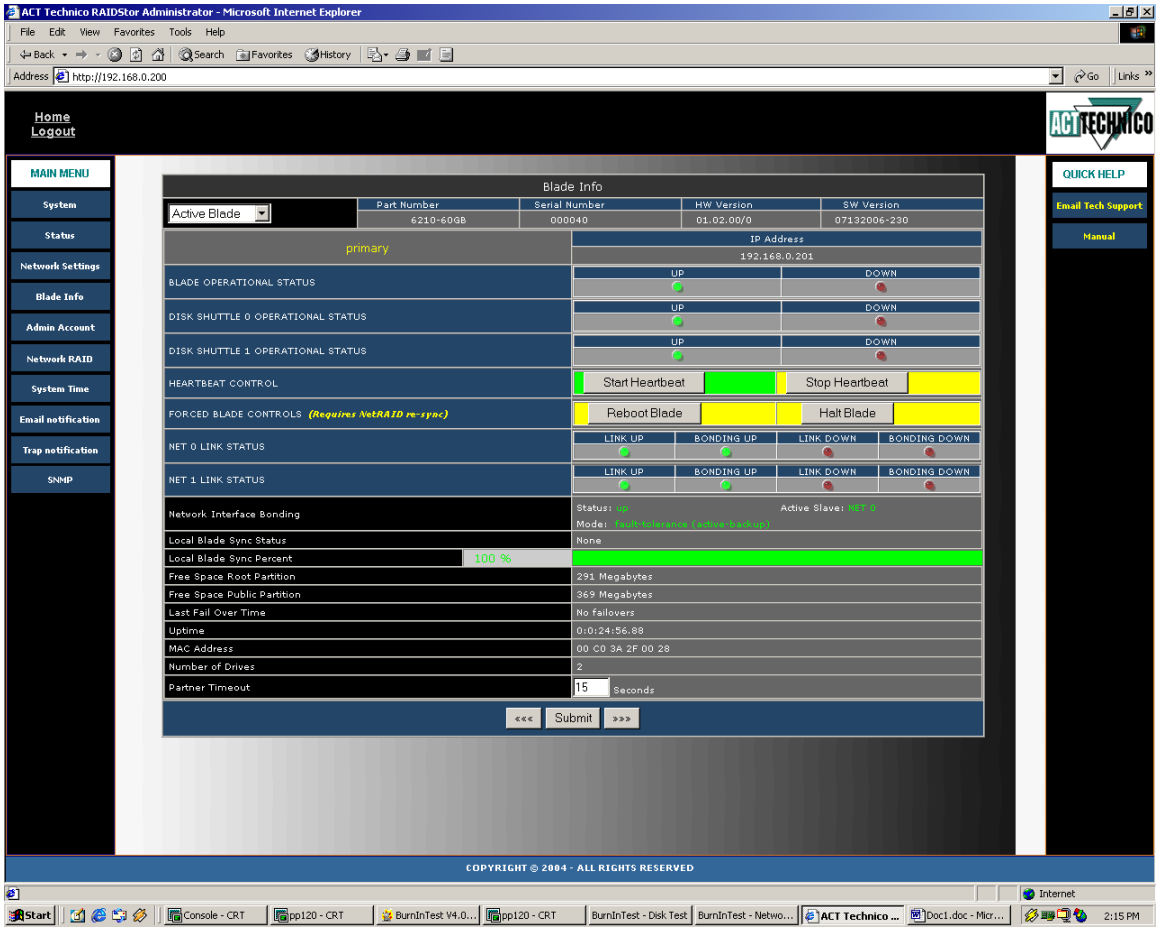


Figure 21: Web Page -- Blade Info -- Dual Blade

4.2.6 Admin Account Page

This page modifies administrative account parameters, specifically the web page login password.

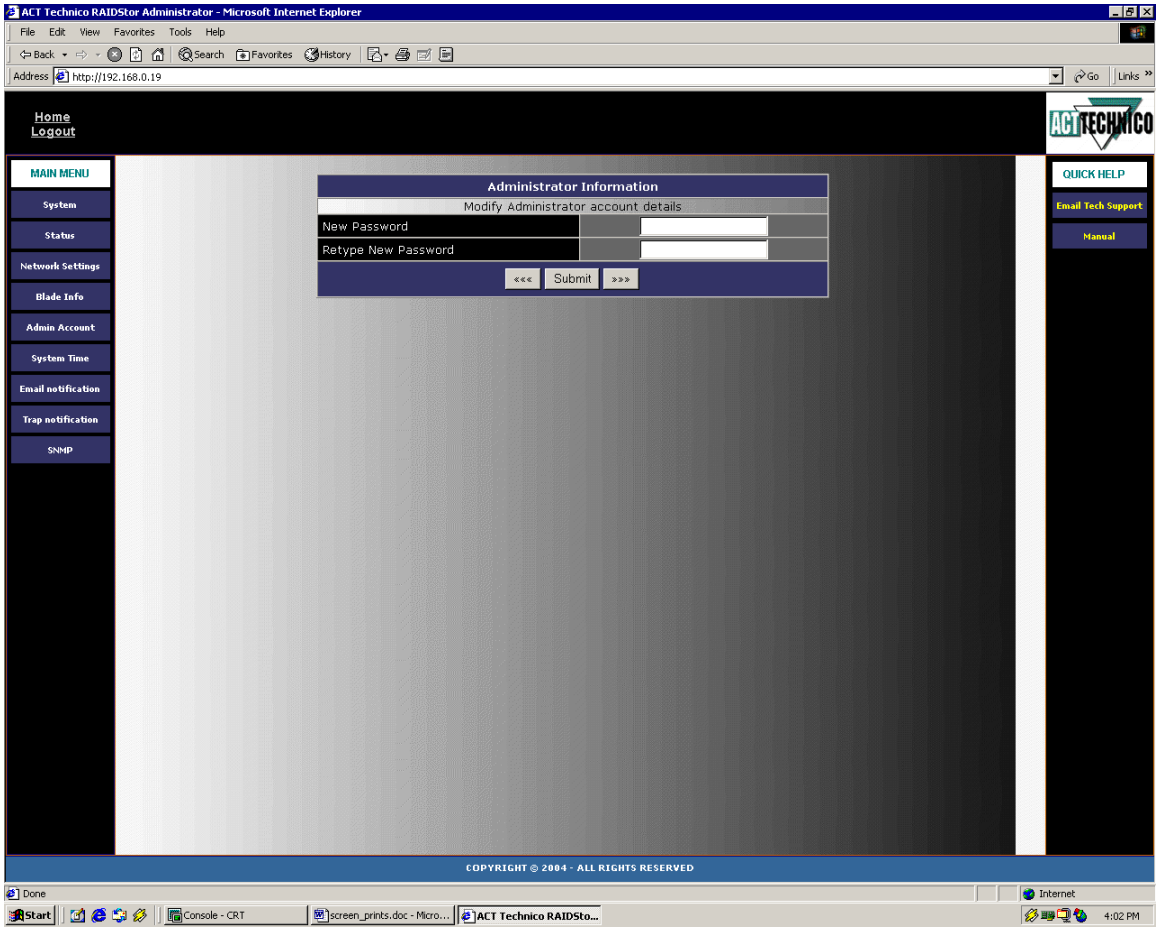


Figure 22: Web Page – Administrative Account

4.2.7 System Time Page

This page displays the date and time.

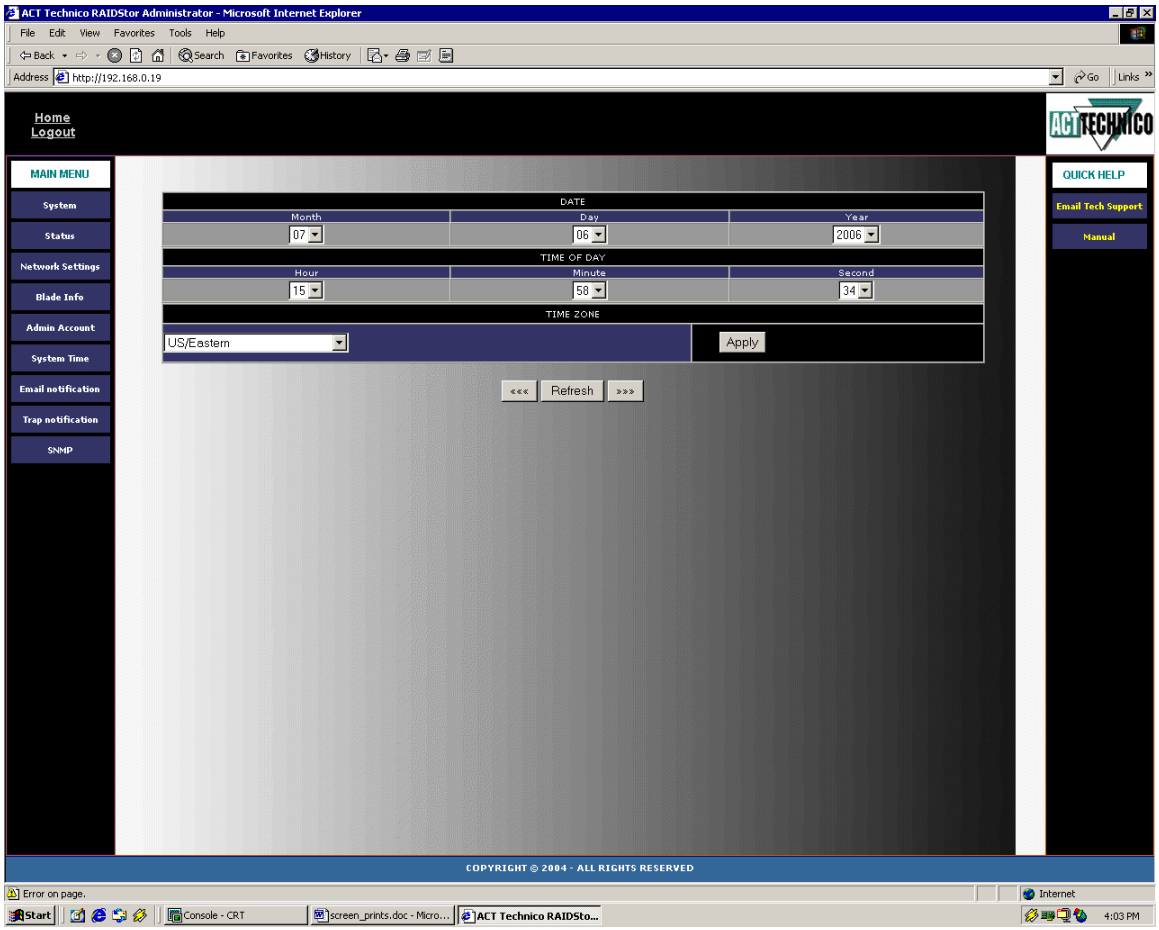


Figure 23: Web Page - System Time

4.2.8 Email Notification Page

This page configures email event notifications. Entries can be added, deleted, or disabled. The list of email addresses is maintained in the RAIDStor config file. Both blades in a dual blade system should have the same values. This synchronization is handled by SNMP.

The SNMP agent used a PERL module to write email messages to the Simple Mail Transfer Port (SMTP). There is no mail application (ie: smail) on the RAIDStor. To use this feature an alias entry needs to be put in the /etc/hosts file for a system named mailhost. SMTP will use this IP as the SMTP destination. For example, to use a system named pp120 for the mail host add thefollowing to /etc/hosts:

10.0.0.1 mailhost pp120

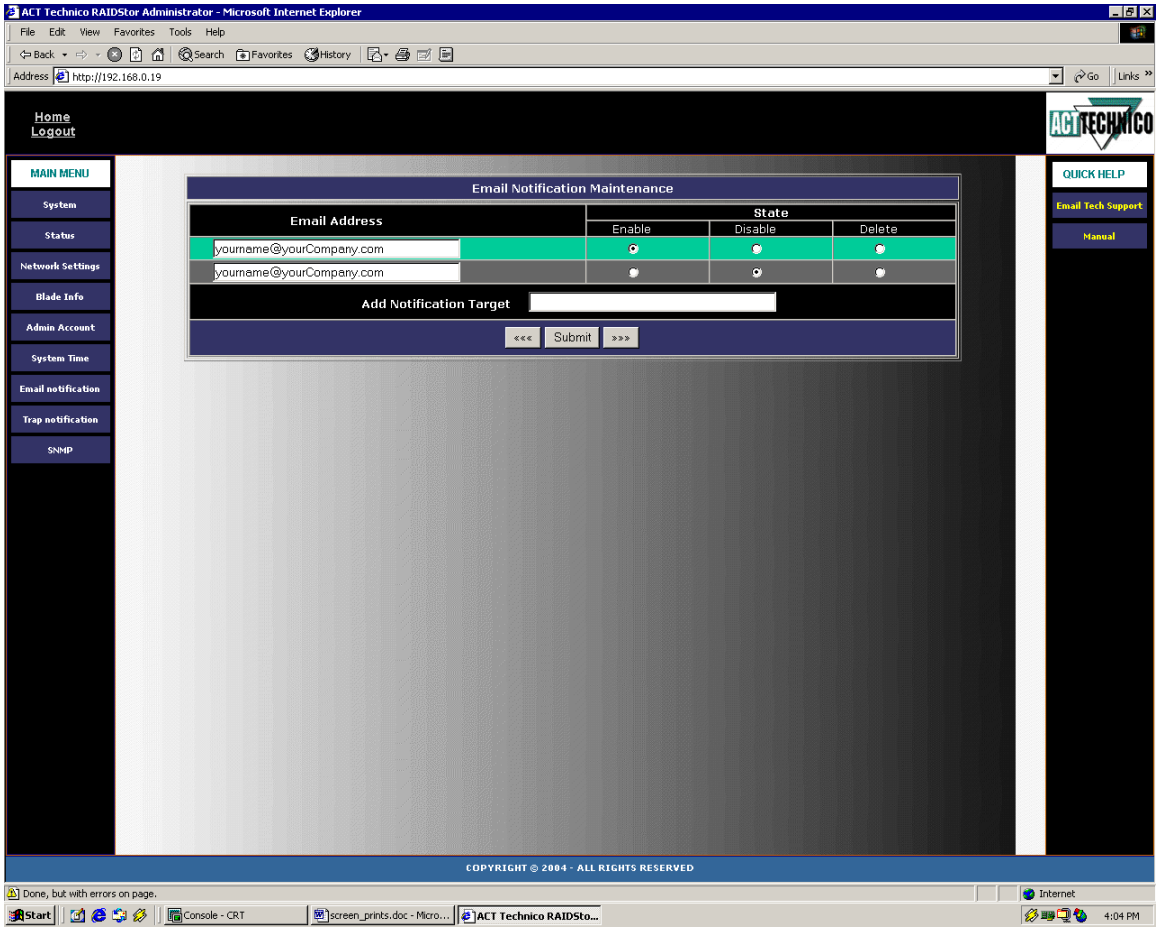


Figure 24: Web Page - Email Notification

4.2.9 Trap Notification Page

This page configures SNMP trap event notification. Entries can be added, deleted, or disabled. The list of traps is maintained in the RAIDStor config file. Both blades in a dual blade system should have the same values. This synchronization is handled by SNMP. Alink to the LOG file of traps is provided.

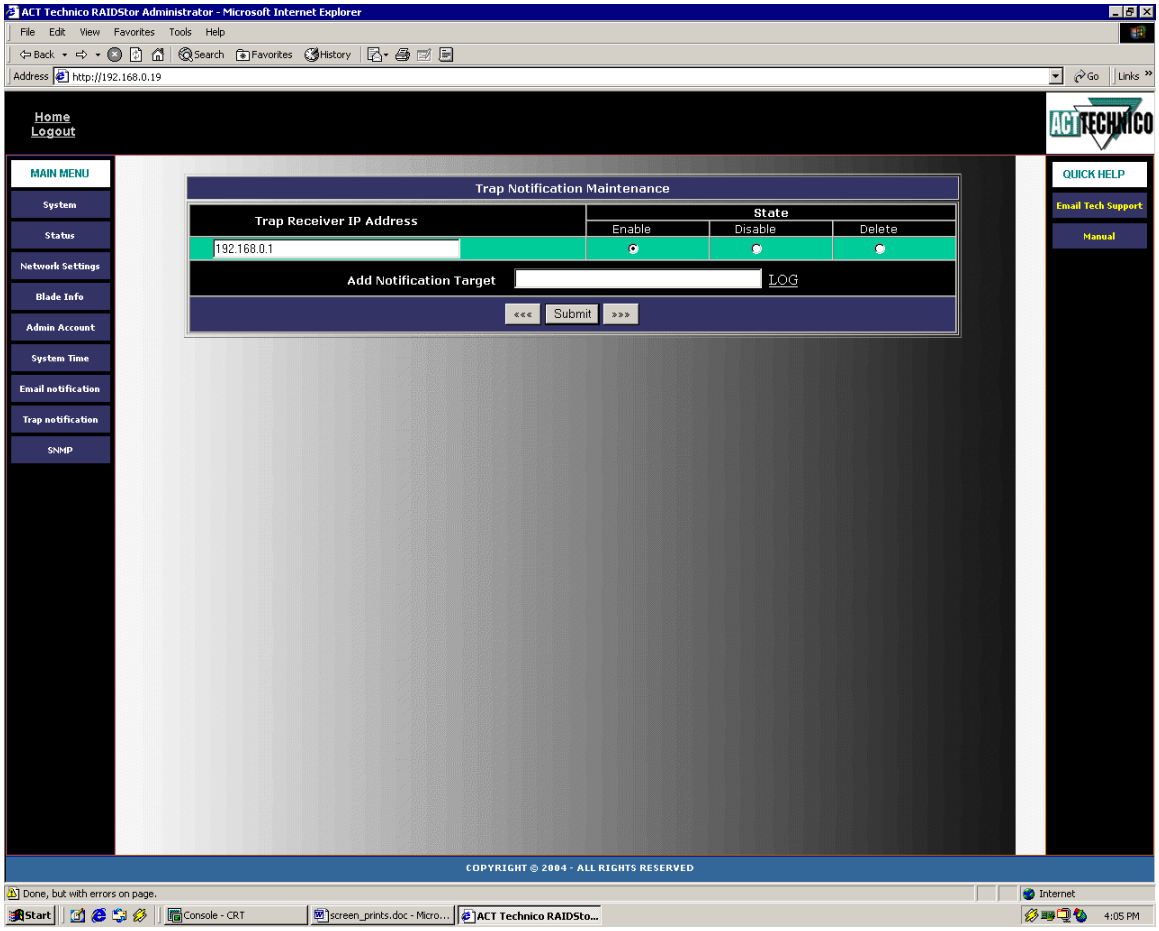


Figure 25: Web Page - Trap Notification

4.2.10 SNMP Web Page

Then SNMP web page displays some basic SNMP stats and allows the user to update the system location and contact.

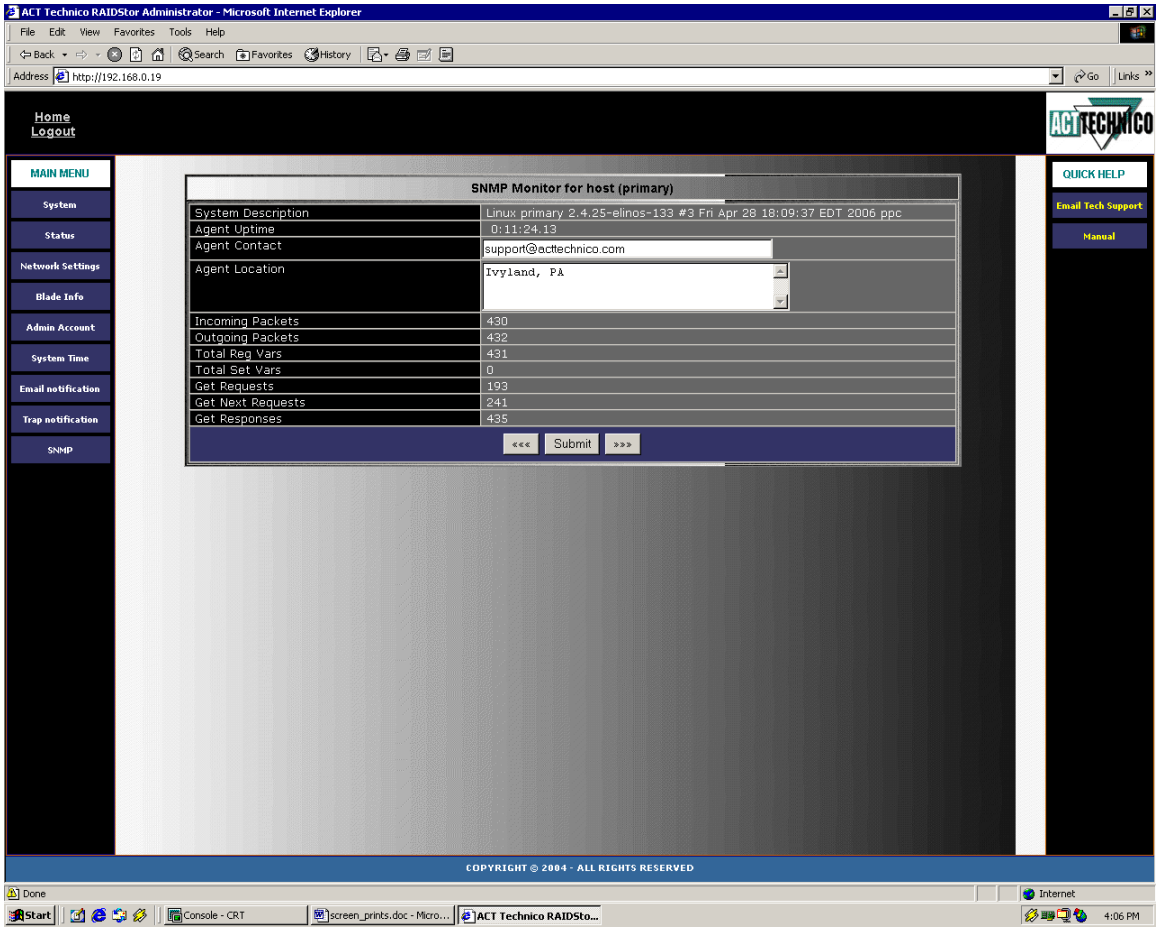


Figure 26: Web Page - SNMP

4.3 SNMP

The Simple Network Management Protocol (SNMP) can be used to remotely monitor and modify the RAIDStor system operation. The SNMP functionality is implemented using the open source net-snmp package (initially version 5.1). The RAIDStor functionality was added to net-smp by extending the default MIB with a section called ACTTECHNICO-RAIDStor-MIB.

4.3.1 SNMP Agent

SNMP get and set calls are issued to the system's IP address (IP address owned by active blade). On a dual blade system, the calls are propagated to the standby blade when appropriate.

4.3.1.1 Normal Operation

Once SNMP startup has completed, the agent begins normal operation. During normal operation, blades monitor their local status and other parameters on a periodic basis (using the alarm thread). The MIB variables are divided into groups of similar values and each group has its own parameter update delay (via the CACHE_CHECK macro). An alarm thread runs through each of these groups with a two second delay between each check. When a group is checked for an update, it compares the current time with the last update and performs the update if that time has expired.

The two-second delay happens after the groups are all updated. This means the actual delay between updates can be much greater than two seconds due to the time it can take to update a parameter. For example, updating the NAS status requires issuing an SNMP get to the remote blade in a dual blade system. This alone can take one or more seconds.

The delay between updates can vary by several seconds due to the delay between updating each group and due to the timing of when current time is compared to the time of the previous update. SNMP requests received from the RAIDStor Web Interface or from a remote SNMP agent can also delay an update due to mutual exclusive protection of lower level SNMP structures. Typically, the update of system status happens every 15-20 seconds, although it may occasionally vary above that range. Some parameters are updated much more slowly (several minutes between updates).

4.3.1.2 SNMP Set Operation

On a single blade RAIDStor SNMP set operations are simply processed as they are received. The SNMP agent code uses a PERL script to determine if the blade is a single blade (/usr/local/bin/rsIsSingleBlade). The script returns 1on a single blade or 0 on a dual blade. A blade configured for run-level 3 is considered a single blade. This item is only checked on SNMP agent startup.

On a dual blade RAIDStor the SBMP agent code keeps both blades in sync. After the SNMP agent has entered normal operations, it can accept get and set calls from the network. A typical sequence of an SNMP set will look like this:

1) The SNMP agent has completed startup and entered normal operation.
2) Access to the Web Interface or a remote SNMP agent causes an SNMP Set command to be transmitted to the RAIDStor virtual IP address.
3) The active blade receives and processes the SNMP set command.
4) If the primary blade is the active blade, the command is forwarded to the backup blade when necessary. Most parameters do require synchronization between blades but the code itself must be consulted to verify whether a specific parameter is affected. Some additional notes on parameter synchronization:
    a) The backup blade will not forward SNMP commands to the primary, even if the backup is the active blade.
    b) The primary blade will not enact a change locally if the parameter cannot be synchronized with the backup blade. For example, if the backup blade is offline no changes can be made via the SNMP interface (SNMP get commands will still work). While the backup blade is down configuration changes would need to be made manually.
5) Once the primary blade successfully synchronizes with the backup, the primary blade enacts the change locally.

4.3.2 Traps and Email Notification

SNMP traps and email notification events are generated asynchronously in response to a change in state or status. The alarm thread discussed in Section 4.3.1.1 (Normal Operation) is responsible for monitoring the state and status values of specific MIB variables. The alarm thread runs on a two second period but, as discussed in Section 4.3.1.1, other factors can significantly increase this delay. See Table 15: SNMP for a list of supported SNMP events.

Trap	Comment
Network RAID Sync Status	This is issued on a sync state change, either starting or stopping of the network RAID sync. If the sync finishes before the alarm task can detect an active synchronization, then the sync will not be reported. This is a dual blade trap only.
NAS State/Status Change	This detects a change in the admin state and operational status of the NAS functionality of the RAIDStor. This status is similar to the System Health Summary discussed in the Web Interface documentation. Keep in mind that state and status here refer to the system level rather than the blade level.
RAIDStor State/Status Change	This detects a change in the admin state and operational status of a RAIDStor blade. Keep in mind that state and status here refer to the blade level rather than the system level.
Network Interface Up/Down	The purpose of this is to alert a user when a network interface goes down and when it comes back up. This is obviously only relevant when at least one network interface is still connected.
Local Disk RAID Sync Status	This is issued on a sync state change, either starting or stopping of the local sync. If the sync finishes before the alarm task can detect an active synchronization, then the sync will not be reported
Disk/RAID Status Change	This is issued in response to a change in disk op status. This refers to hot pluggable disk shuttles. Removal or insertion of a disk shuttle generates a trap.

Table 15: SNMP Traps

4.4 Security

4.4.1 Login Security

The serial port connection is the most secure connection to the RAIDStor command line interface. Telnet and rlogin provide no security, making passwords easily readable by monitoring network traffic. SSH provides encryption making it more difficult to successfully monitor network communications and steal password. For maximum security of the command line interface, ***telnet*** and ***rlogin*** services should be disabled.

4.4.2 DRBD Security

The DRBD driver used on a dual blade RAIDStor, uses unencrypted network protocols and does not employ any security. With DRBD, the complete disk content goes over the network; however, since this data is limited to only the external partition that contains user data, viewing this data will not compromise the security of the RAIDStor unit itself. If user data itself represents a security issue, then it is recommended that both blades be located on a secure network that is accessible only by trusted users.

4.4.3 Server Security

The NFS and SMB/CIFS servers limit access to only the external partition containing user data. There is no access granted to the internal partitions that contain the root file system, so these servers will not compromise the security of the RAIDStor unit itself However, by default no security features are enabled on either the NFS or SMB/CIFS servers and thus any user can read and write any file on the external user data partition.

If user data itself represents a security issue, then it is recommended that the security features in NFS and SMB/CIFS be enabled. For NFS, this consists of adding access restrictions to ***/etc/exports.*** For SMB/CIFS, this consists of adding access restrictions to ***/etc/samba/smb.conf*** and ***/etc/samba/smbpassword***. Please refer to:

http://www.samba.org

for overall Samba server documentation.

4.4.4 Web Page Security

HTTPS (Hypertext Transfer Protocol over Secure Socket Layer, or HTTP over SSL) is a Web protocol developed supported by all modern browsers that encrypts and decrypts user page requests as well as the pages that are returned by the Web server. SSL uses a 40-bit key size for the RC4 stream encryption algorithm, which is considered an adequate degree of encryption for commercial exchange.

The only options for Web Interface communications are the HTTP and HTTPS protocols. HTTPS is a secure protocol that uses encryption while HTTP is insecure. The Apache Web server supports the HTTPS protocol, but the RAIDStor Web Interface does not currently use it nor has it been tested in the current hardware/software configuration. There is no technical restriction preventing the use of HTTPS.

4.4.5 SNMP Security

SNMP security is not implemented in the current SNMP agent.

5 Dual Blade Configuration Procedures using the Command Line

This section explains how to setup and change common settings on a dual blade RAIDStor using the command line. These procedures can be quite complex and are only recommended for experienced users. Please note that is it easy to make erroneous settings when using the command line interface that will render the system inoperable.

All procedures listed assume that Heartbeat and DRBD are running. These services must be shutdown before making changes to the system.

Always start by shutting down Heartbeat on the backup node first, and then shut it down on the primary node. Shutdown on the primary node can begin right after shutdown has been initiated on the backup node. It is not required that shutdown be complete on the backup node before starting shutdown on the primary node.

If Heartbeat is shutdown on the primary node first, a failover to the backup node could occur if the backup node is not shutdown within the heartbeat ***deadtime*** period. If a failover starts, it must be allowed to finish before shut down on the backup node can commence. Also, DRBD may require a re-synchronization once it has been restarted. So, to avoid all of that, just shutdown Heartbeat on the backup node first.

Now that Heartbeat has been shutdown on both blades, all access to the DRBD disks has been disabled. If NFS and/or SMB/CIFS were running, those services have now been stopped. The DRBD disks will no longer be mounted on the primary node (nor on the backup node). However, the DRBD driver is still running and any node-to-node synchronization that is pending can still occur. The DRBD disks will not require re-synchronization once they have been restarted.

The final step is to shutdown the DRBD driver on both blades. Order and timing does not matter.

5.1 Step by Step Shutdown of Heartbeat and DRBD

On the backup blade:

```
heartbeat stop
```

On the primary blade:

```
heartbeat stop
```

Wait for Heartbeat shutdown to complete on both blades before continuing, then on one of the blades:

```
drbd stop
```

On the other blade:

```
drbd stop
```

Wait for DRBD shutdown to complete on both blades before continuing with any other settings changes.

5.2 Step by Step Startup of Heartbeat and DRBD

On one of the blades:

```
drbd start
```

On the other blade:

```
drbd start
```

Wait for DRBD startup to complete on both blades before continuing. This could include a lengthy resynchronization process, which could take several hours.

On the primary blade:

```
heartbeat start
```

On the backup blade:

```
heartbeat start
```

5.3 Change the IP parameters of a RAIDStor Blade From /etc/hosts

The IP address of each RAIDStor blade is specified in the file ***/etc/hosts, /etc/drbd.conf,*** and ***/etc/ha.d/haresources*** on both RAIDStor nodes. The Heartbeat common NAS IP address is contained in the file ***/etc/ha.d/haresources***. Instructions for modifying that address is given later. Although similar, ***/etc/hosts*** is unique for each RAIDStor and may not be copied from node to node.

The IP addresses, common *netmask*, common *timehost*, common *gateway* of both blades are contained in this file. The IP name alias *myraidstor* specifies the IP address of this RAIDStor node. The IP name alias *otherraidstor* specifies the IP address of the other RAIDStor node. *otherraidstor* must be on one of the two address lines for the RAIDStors. The position of *myraidstor* and *otherraidstor* should be the only difference in ***/etc/hosts*** across RAIDStor nodes. All other parameters should be the same. Changing any IP parameter for one node requires an identical change on the other node.

/etc/hosts typically looks like this:

```
127.0.0.1      localhost.localdomain  localhost
192.168.0.1    gateway
192.168.0.200  primary myraidstor
192.168.0.201  backup  otherraidstor
132.163.4.103  timehost
192.168.0.21   mailhost yourMailHostForEmailNotifications
255.255.255.0  netmask
```

Update the primary myraidstor address for a single or primary blade and the backup otherraidstor address for a backup dual blade.

NOTE: The location of *myraidstor* and *otherraidstor*, will be different depending on the RAIDStor node.

The Heartbeat common NAS IP address is contained in the file: ***/etc/ha.d/haresources.*** It is set by the ***IPaddr*** script in that file.

NOTE: For Dual Blade if the Heartbeat common NAS IP is on a different subnet from the *primary* or *backup blade*, then the NAS IP must include the subnet mask in slash format. The default NAS IP is 192.168.0.200, to specify that address with the subnet in slash format use this: 192.168.0.200/24.

Once these files are edited, then the system needs to update the IP address. To update the addresses without rebooting, issue the following commands on both nodes:

```
ifconfig bond0 down
bonding.sh
```

Finally, restart Heartbeat and DRBD as outlined previously if running Dual Blade.

5.4 Add a Service to Heartbeat

Heartbeat is used to start and stop various services that need to be bound to the common shared IP address and run on only one node at a time. These services typically include the DRBD disks, NFS, SMB, and DHCP services. To be able to add a service, the service must have a startup script that can process the *start, stop,* and *status* arguments. These scripts must be located in ***/etc/init.d***. Add the script name to the file ***/etc/ha.d/haresources.*** When heartbeat processes this file, it starts the scripts in the order that they are listed. It is important to have the *IPaddr* script listed first to setup the common shared IP address. Next the DRBD disks are started. After that, the other services are started. Heartbeat stops the scripts in reverse order, with the last listed service shutdown first.

It is important that both nodes have an identical ***haresources*** file. It must be identical down the character or odd heartbeat behaviors will be observed. We have found that using *Webmin* to modify the heartbeat file can sometimes create a slightly different file if it is updated (without any changes) on one node and not on another. If *Webmin* is used to modify a heartbeat services on one node, it should be used to modify the services on the other node to maintain consistency. Alternatively, the ***haresources*** could be copied using FTP to the other node.

5.5 Add a DRBD Disk

Adding a DRBD disk requires modifications to nearly all of the levels of services described previously. If done by hand, the modifications needed to add a DRBD disk will typically take 1-2 hours. ACT/Technico has developed a set of scripts to build a complete RAIDStor system from scratch and it is much easier, quicker, and less error prone to use those scripts to build a brand new system with the desired partitions and DRBD disks than to add a DRBD disk to an existing system.

In summary, the following steps need to be performed:

1. Shutdown Heartbeat and DRBD on both blades
2. Set the run level to 1 on both blades
3. Repartition both disks on both blades
4. Reboot both blades
5. Make a new RAID disk on both blades
6. Modify the DRBD configuration on both blades
7. Modify mount point configuration on both blades
8. Modify NFS or SMB configuration on both blades
9. Modify the Heartbeat configuration on both blades
10. Set the run level to 4 on both blades
11. Start Bonding Driver and DRBD on both blades
12. Create a filesystem on the DRBD disk
13. Start Heartbeat on both blades
14. Set access permissions

The details of each step are shown next.

5.5.1 Shutdown Heartbeat and DRBD

Using the procedure previously shown, stop Heartbeat and DRBD on both blades.

5.5.2 Change the Run Level

Set the Linux run level to 1. To do this, edit the file ***/etc/inittab***. Change the following line:

```
id:4:initdefault:
```

to:

```
id:1:initdefault:
```

Make sure that all settings are identical on both blades.

5.5.3 Repartition the Disks

If there is no free space on the disks and a current partition or partitions need to be deleted, then any RAID(s) that are running on those partitions need to be stopped. To determine which RAID(s) are running on which partition(s), use the command ***pmd*** (or ***more /proc/mdstat*****).** That command shows each RAID disk device and what partitions it uses. To stop the RAID(s), issue one or more of the following command on both blades:

```
raidstop /dev/md/{RAID device number to be stopped}
```

Now repartition both disks on both blades. This means that you need to run ***fdisk*** four or eight times and setup up identical partitions on all disks. Use the following command for the first disk (*host0* is the first disk and *host1* the second):

```
cfdisk /dev/scsi/host0/bus0/target0/lun0/disc
```

Once the partition has been created, its type needs to be set to *Linux raid autodetect*. Set the type to ***fd***.

When all partitions have been created, use the ***w*** command to write the partition table and exit. You will see the following message:

```
WARNING: Re-reading the partition table failed with error 16: Device or
resource busy.
The kernel still uses the old table.
The new table will be used at the next reboot.
```

This message is normal and is the reason the blades must rebooted after partitioning all disks.

Now set up identical partitions on the other disk on both blades. Use the following command to setup the second disk on a blade:

```
fdisk /dev/scsi/host1/bus0/target0/lun0/disc
```

Make sure that all settings are identical on both disks on both blades.

Caution: do not change either of the RFS partitions or the RAIDStor may become unbootable!

5.5.4 Reboot the Blades

Use the ***reboot*** command to restart the blades. Once the system has booted, login.

5.5.5 Make a New RAID

Edit the file ***/etc/raidtab*** on both blades by adding a new RAID entry. An example RAID entry looks like this:

```
raiddev /dev/md/{RAID device number}
  raid-level  0
  nr-raid-disks 2
  chunk-size  32
  persistent-superblock 1
  device  /dev/scsi/host0/bus0/target0/lun0/part{partition number}
  raid-disk 0
  device  /dev/scsi/host1/bus0/target0/lun0/part{partition number}
  raid-disk 1
```

Make sure that all settings are identical on both blades.

Create the RAID by using this command (on both blades):

```
mkraid /dev/md/{RAID device number}
```

To verify that the RAID is running, use the ***pmd*** (or ***more /proc/mdstat)*** command to get a status listing of the RAIDs.

5.5.6 Modify the DRBD Configuration

Edit the file ***/etc/drbd.conf*** on both blades by adding a new DRBD disk entry. An example DRBD disk entry looks like this:

```
resource {DRBD disk name} {
...

 on primary {
  device = /dev/drbd/{DRBD device number}
  disk  = /dev/md/{RAID device number}
  address = primary:{7788 + DRBD device number}
  meta-disk  = /dev/md/2 [index]
 }

 on backup {
  device = /dev/drbd/{DRBD device number}
  disk  = /dev/md/{RAID device number}
```

```
  address = backup:{7788 + DRBD device number}
  meta-disk  = /dev/md/2 [index]
 }
}
```

Other parameters that need to be adjusted are *resource, device, disk, address:port, and meta-disk.* These need to be modified for both hosts listed in the DRBD device entry. The *resource* parameter needs to be set to the new disk's name. The *device* parameter needs to be set to an unused DRBD device number; typically it will be the next unused number. The *disk* parameter needs to be set to the newly created RAID device. The *port* parameter needs to be set to an unused port number; typically it will be set to 7788 + DRBD device number.

Make sure that all settings are identical on both blades.

5.5.7 Modify mountpoint Configuration

Add a mount point for the new DRBD device. It should be located in the root directory and named the same as the name specified by the *resource* parameter in ***/etc/drbd.conf***. Execute the following command:

```
mkdir /{DRBD disk name}
```

Optionally set the access permissions. If the new DRBD disk is to be publicly readable and writable, then set the permissions to this:

```
chmod 777 /{DRBD disk name}
```

Notes: this command only changes the permissions of the mount point, not the mounted DRBD disk and this command will need to be executed again when the DRBD disk is mounted.

Now add a *Filesystem* command to ***/etc/ha.d/haresource*** create a mount point entry so that Heartbeat can mount and unmount the DRBD disk. An example mount point entry looks like this:

```
Filesystem::/dev/drbd/{DRBD device number}::/{DRBD disk name}::ext3
```

Heartbeat is responsible for mounting and unmounting the DRBD disks.

Make sure that all settings are identical on both blades.

5.5.8 Modify NFS or SMB Configuration

This step is only required if NFS or SMB access to the DRBD disk is needed.

To add NFS access to the DRBD disk, add an export entry to the file ***/etc/exports.*** An example export entry looks like this:

```
/{DRBD disk name}  *(rw,sync,no_wdelay,all_squash,anonuid=-2,anongid=-2)
```

Note: the *sync* and *no_wdelay* options are needed to provide better fault tolerance characteristics. See a previous section for details. The *all_squash* options force all files to be owned by *user* and *group* to a –2. This is the default user and group used by *Windows Services for UNIX.*

To add SMB access to the DRBD disk, add an entry to the file ***/etc/samba/smb.conf***. An example SMB entry looks like this:

```
[{SMB share name}]
  path = /{DRBD disk name}
  read only = no
  guest ok = yes
  force user = nobody
  force group = nobody
```

This is an entry for anonymous (available to all) access.

Make sure that all settings are identical on both blades.

5.5.9 Modify the Heartbeat Configuration

Heartbeat needs to know about the new DRBD disk so that it can mount it and unmount it. To do this, edit the file ***/etc/ha.d/haresouces.*** This file contains a single line that has the script names and arguments for each service to be controlled. The first script is always ***IPaddr.*** After that, the DRBD disk scripts are given. There is one script for each DRBD disk. Add an entry to the ***haresources*** file, after the other DRBD disk scripts, but before any other services, that looks like this:

```
drbddisk::{DRBD disk name}
```

For example, a DRBD disk entry in ***haresources*** would look similar to this:

```
primary IPaddr::192.168.0.200 drbddisk::{DRBD disk name} {other services}
```

5.5.10 Change the Run Level

Set the linux run level back to 4. To do this edit the file ***/etc/inittab***. Change the following line:

```
id:1:initdefault:
```

to:

```
id:4:initdefault:
```

Make sure that all settings are identical on both blades.

5.5.11 Start Bonding Driver and DRBD

First start the bonding driver. To do this, execute the following command on both blades:

```
bonding.sh
```

Next start DRBD. To do this, execute the following command on one blade:

```
drbd start
```

First, start DRBD on only one blade and check for errors. If any errors are found, DRBD will not be started. Correct the error(s) and try starting DRBD again. Then start up DRBD on the other blade and again check for errors.

5.5.12 Create a Filesystem on the DRBD disk

Create and tune a filesystem on the DRBD disk by executing these commands on one of the blades (but not both):

```
drbdsetup /dev/drbd/{DRBD device number} primary
mkfs.ext3 /dev/drbd/{DRBD device number}
tune2fs –c 1 –i 0 /dev/drbd/{DRBD device number}
```

Note: since we are working with DRBD devices, we only need to run these commands on one of the blades. DRBD will automatically replicate all changes to the other node.

5.5.13 Start Heartbeat

Finally start Heartbeat. To do this, execute the following command on one blade:

```
heartbeat start
```

First, start Heartbeat on only one blade and check for errors. If any errors are found, Heartbeat will not be started. Correct the error(s) and try starting Heartbeat again. Once Heartbeat is started on one blade, then start it up on the other blade within 120 seconds and again check for errors.

Check for DRBD disk mount problems by executing the command ***pdr*** (***cat /proc/drbd***) on the primary blade. Note that it takes about ½ a minute to mount all DRBD disks and set them in their primary state. ***pdr*** returns output similar to this:

```
0: cs:Connected st:Primary/Secondary ns:12088 nr:0 dw:12088 dr:103085 pe:0 ua:0
```

```
1: cs:Connected st:Primary/Secondary ns:8008 nr:0 dw:8008 dr:68373 pe:0 ua:0
2: cs:Connected st:Primary/Secondary ns:0 nr:0 dw:0 dr:45 pe:0 ua:0
3: cs:Connected st:Primary/Secondary ns:0 nr:0 dw:0 dr:45 pe:0 ua:0
```

5.5.14 Set Access Permissions

Optionally set the access permissions. If the new DRBD disk is to be publicly readable and writable, then set the permissions to this:

```
chmod 777 /{DRBD disk name}
```

Note: this changes the permissions of the mounted DRBD disk. This command should only be run on the primary blade with the DRBD disk mounted.

6 RAIDStor Software Installation

ACT has an automated procedure to rebuild a RAIDStor that will partition the hard drives, make all necessary filesystems and copy software to the RAIDStor across the network. This procedure will **DESTROY** all original data on the blade. The user must perform the following tasks:

- A serial console session is established with the RAIDStor.
- A capable host is chosen to be the DHCP and NFS server for rebuilding the RAIDStor.
- The DHCP server is setup with the latest software.
- The MAC address of the RAIDStor is noted.
- The DHCP server is configured to supply the Linux kernel to the RAIDStor with the MAC address noted above.
- The firmware is configured to retrieve the blade IP address from a DHCP server and mount the /root filesystem over the network
- The user boots the blade with the firmware "**nboot**" command
- The user issues the "**installrfs**" command with the appropriate options.

6.1 DHCP Server Setup

- The software is provided as a tar.gz file named **rs_###.tgz**, where ### is the release number. The software release for a RAIDStor can be determined from the **Blade Info** screen on the web interface or by running the following command from the /usr/share/snmp directory of an installed RAIDStor:

  ```
  cat raidstor.cfg | grep rsSoftwareVersion
  ```

- To install and configure a RAIDStor from scratch, this file should be un-tarred into a directory named /raidstor on the DHCP server.
- The /raidstor directory should be exported via NFS with the root_no_squash option. The /raidstor entry in the /etc/exports file should look like the following.
  - /raidstor *(rw,sync,no_wdelay,no_root_squash)
- In the top level of the /raidstor directory there is a /**tftpboot** directory. The Linux kernel booted by the RAIDStor is stored in this directory. The file is named pImage. This file should be copied to a tftpserver so that it can be retrieved on boot. Copy pImage to the /tftpboot directory of the DHCP server.

6.1.1 Record the MAC address of Target RAIDStor

Connect a RAIDStor serial cable to the console port on the RAIDStor to be installed and monitor the serial console on boot. The following screen will be seen:

```
_____________________ Secondary MenMon for the D303 Version 0.95 _____________
|                                                                             |
|              Parts of this code based on Motorola's Dink32                  |
|                   Created  Oct 24 2005     10:36:54                         |
|_____________________________________________________________________________|
|    HW Revision: 01.01.00            |            CPU: MPC8245     (MAP B)    |
|  Serial Number: 19                  |  CPU/MEM Clock: 400 / 132.0 MHz        |
|    Board Model: D303-00             |    DIMM Module: 256 MB Setup: 222      |
|  CPLD Revision: 0x07                |                                        |
 \____________________________________________________________________________/
```

```
press 'ESC' to setup/MENMON
Shuttle 0: PRESENT
Shuttle 1: PRESENT
Shuttle 2: PRESENT
Shuttle 3: PRESENT
```

Immediately upon booting Hit the **ESC** key to stop auto-boot to prompt Menmon. If the Menmon prompt was missed hit the Reset button on the RAIDStor to reset the system.

The number of shuttles present will vary depending on blade configuration. Note the Serial Number of the RAIDStor blade, this will be used to setup DHCP. The serial number must be converted to hex to configure DHCP. Another option is issuing the nboot command from Menmon and the hex value of the serial number can be seen at the end of the address.

6.1.2 DHCP Configuration

There are two types of RAIDStor blades, single blade and dual blade. A single blade RAIDStor is always configured as the "**primary**" blade. When a dual blade RAIDStor is configured one blade is configured as the **primary** and one as the **backup**. The primary and backup blades are passed different names and mount a different root-path.

6.1.2.1 Single Blade or Primary Dual Blade DHCP Setup

Create the following entry in the /etc/dhcpd.conf file on the DHCP server.

```
host d303_pri_PP {
        hardware ethernet 00:c0:3a:2f:00:13;
        fixed-address 192.168.0.201;
        server-name "192.168.0.30";
        option root-path "/raidstor/rfs/d303_pri";
        option host-name "primary";
        filename "pImage";
    }
```

- Change the last digit of the hardware Ethernet address (highlighted above) to match the hex serial number of the RAIDStor blade being rebuilt/installed. Numbers above 255 will alter the last two digits. Note, the serial number is displayed in decimal on boot.
- Set the **server-name** field to the IP address of the DHCP server.
- Set the **fixed-address** field to the IP address designated for the target RAIDStor, units are shipped with 192.168.0.201 as the default for primary blades.
- The subnet, netmask, and other fields should be setup as desired by the user.
- Restart the DHCP server so these changes take effect.

6.1.2.2 Backup Blade DHCP Setup

The second blade (the backup) in a dual blade RAIDStor is setup as follows:

```
host d303_sec_BB {
        hardware ethernet 00:c0:3a:2f:00:28;
        fixed-address 192.168.0.202;
        server-name "192.168.0.30";
        option root-path "/raidstor/rfs/d303_sec";
        option host-name "backup";
        filename "pImage";
    }
```

- Change the last digit of the hardware Ethernet address (highlighted above) to match the hex serial number of the RAIDStor blade being rebuilt/installed. Numbers above 255 will alter the last two digits. Note, the serial number is displayed in decimal on boot.
- Set the **server-name** field to the IP address of the DHCP server.
- Set the **fixed-address** field to the IP address designated for the target RAIDStor, units are shipped with 192.168.0.201 as the default for primary blades.
- The subnet, netmask, and other fields should be setup as desired by the user.
- Restart the DHCP server so these changes take effect.

6.1.3 MenMon Configuration

Configuration parameters are stored in an EEPROM. The default startup options needs to be set to "dboot" to boot from the disk. The "kerpar" parameter needs to be set to tell the kernel where to find the root filesystem. Initially the /root filesystem is set to /dev/nfs during the install. The install procedure accesses the EEPROM during the install procedure and sets it to /dev/md/0 to boot from the first raid partition on the hard drive. Reset the board when complete to save the changes.

- From the MenMon> prompt set the startup option to dboot

  MenMon> **ee-startup dboot**

- From the MenMon> prompt set the **kerpar** option to use bootp and nfs

  MenMon> **ee-kerpar ip=bootp root=/dev/nfs rw raid=noautodetect**

6.2 Installing the Software

At this point, the RAIDStor is setup to allow software installation. There are 8 possible software configurations to load, 4 for single blade and 4 for dual blade. Table 1 below lists all of the possible configurations and the command used to install the software.

Single/Dual	# Shuttles	RAID	Command
Single	2	0	installrfs act prod single raid0 2shut
Single	2	1	installrfs act prod single raid1
Single	4	5	installrfs act prod single raid5
Single	4	0	installrfs act prod single raid0 4shut
Dual	2	1+1	installrfs act prod 1+1
Dual	2	1+0	installrfs act prod 1+0 2shut
Dual	4	1+0	installrfs act prod 1+0 4shut
Dual	4	1+5	installrfs act prod 1+5

Table 16: RAIDStor Installation Command Table

- From the MenMon> prompt issue the network boot command (nboot)

  MenMon> nboot

- When the login prompt appears login as **root**, there is no password.
- The date must be set correctly for the software to install correctly

  # date mmddhhmmyyyy (month,day,hour,minute,year)

- Save the date to the RTC with the hwclock command

  # hwclock --systohc

- **Prior** to starting the install, edit the installrfs script in the top-level directory of the target RAIDStor and set the "**hostip"** address to the address of the NFS server. Lookup the software install command in Table 1 above and issue it to start the install.

  # installrfs act prod single raid5

The installrfs command used is configuration dependent. The example above is for a single blade RAID-5 configuration. On a dual blade RAIDStor, the same command is used for both blades. The backup blade receives a slightly different software load because it uses a different root filesystem than the primary blade. This was setup by the DHCP configuration above. When the installation is complete the RAIDStor will reboot.

When the installation is complete the RAIDStor will automatically reboot from the hard drive. The installation procedure automatically updates the firmware to configure the root filesystem to be /dev/md/0.

6.3 Post Installation Procedure

As a result of the software installation, all user configurations and data have been erased. The RAIDStor has been returned to the factory default state. At this point the RAIDStor is ready to be configured and returned to service.

7 RAIDStor Sustained NFS Performance

- There are 8 standard RAIDStor configurations. The table below lists the sustained read/write performance ACT has measured across NFS using an Intel PentiumM host running Linux with a 2.6.18.6 kernel. A number of structured tests were run to determine if performance could be tuned. It was determined that the only change that could be made that would have a measurable effect on performance was stopping the SNMP service. Read/write block size did not have a material effect on performance for block sizes 4K and above. The numbers below were all taken from test runs using 16K bytes for the block size to be consistent. In addition, the mount command on the PentiumM host card was passed the options rsize=32768,wsize=32768.

Sustained NFS Performance (MB/sec)					
Single/Dual	**# Shuttles**	**RAID**	**Write**	**Read**	**Configuration**
Single	2	0	6.72	9.07	All Services Running
Single	2	0	8.52	11.50	No SNMP, No Web Server
Single	2	1	6.27	8.71	All Services Running
Single	2	1	7.80	10.31	No SNMP, No Web Server
Single	4	5	5.28	8.48	All Services Running
Single	4	5	6.82	10.97	No SNMP, No Web Server
Single	4	0	6.76	8.90	All Services Running
Single	4	0	8.63	11.49	No SNMP, No Web Server
Dual	2	1+1	3.42	7.74	All Services Running
Dual	2	1+1	4.69	9.35	No SNMP, No Web Server
Dual	2	1+0	3.39	7.62	All Services Running
Dual	2	1+0	4.76	9.44	No SNMP, No Web Server
Dual	4	1+0	3.75	7.62	All Services Running
Dual	4	1+0	4.92	9.50	No SNMP, No Web Server
Dual	4	1+5	2.85	7.0	All Services Running
Dual	4	1+5	4.02	8.92	No SNMP, No Web Server

Table 17: Sustained NFS Performance

- **There is NO NFS performance difference between the 6210 and 6211 Shuttles**.

8 Environmental, Safety and Reliability Specifications

8.1 Power

Power is supplied to the local logic via the CompactPCI J1 connector. The necessary backplane supplied voltages are +5V and +3.3V. Power is supplied to the disk shuttles via the P26/P27 connectors. Only +5V is supplied to the shuttles, so only +5V hard drives are supported on the RAIDStor. +12V and –12V are not used by the RAIDStor.

- 5V±5%, @ 3.5A
- 3.3V±5%, @ 2.6A.

Current numbers are estimates including two RAIDStor shuttles and hard drives. The actual number will vary with the hard disk selected.

8.2 Safety

- ALL PCBs are manufactured with a flammability rating of 94V-0 by UL recognized manufacturers

8.3 Environmental Specifications RAIDStor Baseboard Only – NO Drives

- Temperature range (operation):
  - +5 to +55°C
  - Airflow: min. 10m³/h
- Temperature range (storage): -40 to +85°C
- Relative humidity (operation): max. 95% non-condensing
- Relative humidity (storage): max. 95% non-condensing
- Altitude: -300m to + 3,000m
- Shock: 10g/11ms
- Vibration (sinusoidal): 2g/10..150Hz

Actual environmental specifications will be affected by the hard drives installed on the RAIDStor.

8.4 Reliability

The RAIDStor consists of a number of components. The MTBF numbers presented in this section are for individual RAIDStor components and several sample configurations. The RAIDStor is available in single blade and dual blade configurations. In a dual blade configuration all components have at least one backup component.

The RAIDStor is intended to provide network mountable storage to clients over the Ethernet. The RAIDStor is considered to be operational as long as network clients machines can mount and use this storage.

In a single blade configuration redundant Ethernet ports are available. Failure of a single Ethernet port will not cause a RAIDStor to become non-operational. In a single blade configuration with two drives configured as RAID1, a redundant disk shuttle and hard drive is available. In this configuration, failure of a single drive or disk shuttle will not cause the RAIDStor to become non-operational. In a single drive configuration with four disk shuttles and four hard drives configured for RAID5, failure of a single drive will not cause the RAIDStor to become non-functional. In a single blade RAIDStor configured for RAID0, failure of a single hard drive will cause the RAIDStor to become non-functional.

In a dual blade configuration the public storage services are only visible from one of the RAIDStors at any given time. The other RAIDStor keeps an up-to-date copy of all of the user data in the public storage. A dual blade RAIDStor pair can continue to service network storage requests if a single blade fails completely and one of the remaining Ethernet ports also fails. If the blade is configured for RAID1 (two drives per blade) or RAID5 (four drives per blade) it can also tolerate an additional hard drive failure or disk shuttle failure.

8.4.1 MTBF – MIL-STD-217 Ground Benign @ 40°C

The component calculations here do not take redundancy into account.

- RAIDStor baseboard: 110,958 hours
- Disk Shuttle (no drives): 2,820,874 hours
- Hard Drive Expansion Board (no drives): 1,766,784 hours
- DVD Expansion Board (no drives): 2,925,170 hours
- Rear Transition Module (RTM): 1,166,350 hours

The following configurations do not take redundancy into account because the drives are not included.

- RAIDStor + 4 disk shuttles + hard drive expansion + RTM + DVD Expansion (no drives)
  - 81,800 hours
- RAIDStor baseboard with two Disk Shuttles (no drives)
  - 102,600 hours

The following is a sample configuration that does not take redundancy into account.

- Typical usage rating for industrial extended temperature hard drives is 300,000 hours
- Typical DVD usage rating is 60,000 hours
- Sample configurations assuming the above ratings for HDD and DVD drives:

- Single Blade (No Redundancy – RAID 0)

  - RAIDStor baseboard, 2 Shuttles, 2 hard drives: 60,910 hours
  - RAIDStor baseboard, hard drive expansion board, 4 Shuttles, 4 hard drives: 39,130 hours

9 Mechanical Specifications – Connector Pin Assignments

9.1 Board Mechanics

- The board conforms to the CompactPCI specification.
- Dimensions: standard double Eurocard, 233.3mm x 160mm
- Weight (without accessories): 380g

9.1.1 CD-ROM Connector P25

The RAIDStor supports the following pin assignment of the female 44-pin plug connector for connecting a CD-ROM:

1	/SIDE-RST	2	GND
3	SIDE-D7	4	SIDE-D8
5	SIDE-D6	6	SIDE-D9
7	SIDE-D5	8	SIDE-D10
9	SIDE-D4	10	SIDE-D11
11	SIDE-D3	12	SIDE-D12
13	SIDE-D2	14	SIDE-D13
15	SIDE-D1	16	SIDE-D14
17	SIDE-D0	18	SIDE-D15
19	GND	20	-
21	SIDE-DRQ	22	GND
23	/SIDE-WR	24	GND
25	/SIDE-RD	26	GND
27	SIDE-RDY	28	-
29	/SIDE-DAK	30	GND
31	SIDE-IRQ	32	-
33	SIDE-DA[1]	34	-
35	SIDE-DA[0]	36	SIDE-DA2
37	/SIDE-CS[0]	38	/SIDE-CS[1]
39	-	40	GND
41	+5V	42	+5V (Motor)
43	GND	44	Reserved

Table 18: Pin Assignment of the 44-Pin CD-ROM Plug Connector P25

Signal	Direction	Function
/SIDE_RST	Out	Reset
SIDE_D15.. SIDE_D0	In/out	Data
SIDE_DMARQ	In	DMA request
/SIDE_DMACK	Out	DMA acknowledge
/SIDE_WR	Out	IO write signal
/SIDE_RD	Out	IO read signal
/SIDE_RDY	In	IO ready
SIDE_A2.. SIDE_A0	Out	Addresses
/SIDE_CS1.. /SIDE_CS0	Out	Chip selects
SIDE_IRQ	In	Interrupt request
GND		Ground

Table 19: Signal Mnemonics of 44-pin Plug Connector for CD-ROM

9.1.2 CD-ROM Adapter

A CD-ROM drive may be adapted via a CD-ROM Adapter. The adapter has two connectors:

9.1.2.1 Connector P24

1	/SIDE-RST	2	GND
3	SIDE-D7	4	SIDE-D8
5	SIDE-D6	6	SIDE-D9
7	SIDE-D5	8	SIDE-D10
9	SIDE-D4	10	SIDE-D11
11	SIDE-D3	12	SIDE-D12
13	SIDE-D2	14	SIDE-D13
15	SIDE-D1	16	SIDE-D14
17	SIDE-D0	18	SIDE-D15
19	GND	20	-
21	SIDE-DRQ	22	GND
23	/SIDE-WR	24	GND
25	/SIDE-RD	26	GND
27	SIDE-RDY	28	-
29	/SIDE-DAK	30	GND
31	SIDE-IRQ	32	-
33	SIDE-DA[1]	34	-
35	SIDE-DA[0]	36	SIDE-DA2
37	/SIDE-CS[0]	38	/SIDE-CS[1]
39	-	40	GND
41	+5V	42	+5V (Motor)
43	GND	44	Reserved

Table 20: Pin Assignment of the 44-Pin CD-ROM Plug Connector P24

9.1.2.2 Connector J4

1	-	2	-
3	-	4	
5	/SIDE-RST	6	SIDE-D8
7	SIDE-D7	8	SIDE-D9
9	SIDE-D6	10	SIDE-D10
11	SIDE-D5	12	SIDE-D11
13	SIDE-D4	14	SIDE-D12
15	SIDE-D3	16	SIDE-D13
17	SIDE-D2	18	SIDE-D14
19	SIDE-D1	20	SIDE-D15
21	SIDE-D0	22	SIDE-DRQ
23	GND	24	/SIDE-RD
25	/SIDE-WR	26	GND
27	SIDE-RDY	28	/SIDE-DAK
29	SIDE-IRQ	30	-
31	SIDE-DA[1]	32	-
33	SIDE-DA[0]	34	SIDE-DA2
35	/SIDE-CS[0]	36	/SIDE-CS[1]
37	-	38	+5V
39	+5V	40	+5V
41	+5V	42	+5V
43	GND	44	GND
45	GND	46	GND
47	-	48	GND
49		50	

Table 21: Pin Assignment of the 50-Pin CD-ROM Connector

9.1.3 CompactFlash Connector P28

The RAIDStor supports the following pin assignment of the female 50-pin plug connector for connecting a CompactFlash:

1	GND	26	-
2	PIDE_D[3]	27	PIDE_D[11]
3	PIDE_D[4]	28	PIDE_D[12]
4	PIDE_D[5]	29	PIDE_D[13]
5	PIDE_D[6]	30	PIDE_D[14]
6	PIDE_D[7]	31	PIDE_D[15]
7	/PIDE_CS1	32	/PIDE_CS3
8	PIDE_A[10]	33	-
9	/ATASEL	34	/PIDE_RD
10	PIDE_A[9]	35	/PIDE_WR
11	PIDE_A[8]	36	/WE
12	PIDE_A[7]	37	PIDE_IRQ
13	3.3V	38	3.3V
14	PIDE_A[6]	39	/CSEL
15	PIDE_A[5]	40	-
16	PIDE_A[4]	41	/PIDE_RST
17	PIDE_A[3]	42	PIDE_RDY
18	PIDE_A[2]	43	-
19	PIDE_A[1]	44	/REG
20	PIDE_A[0]	45	-
21	PIDE_D[0]	46	-
22	PIDE_D[1]	47	PIDE_D[8]
23	PIDE_D[2]	48	PIDE_D[9]
24	-	49	PIDE_D[10]
25	-	50	GND

Table 22: Pin Assignment of the 50-pin CompactFlash Connector

9.1.4 Disk Shuttle Carrier Connector J6

The RAIDStor supports a connector for connecting the disk shuttle carrier board to the RAIDStor. The connector is a 2x20 High-precision female header.

A1	GND	B1	GND
A2	RX_P2	B2	GND
A3	RX_N2	B3	GND
A4	GND	B4	GND
A5	TX_P2	B5	GND
A6	TX_N2	B6	GND
A7	GND	B7	GND
A8	GND	B8	RX_P3
A9	GND	B9	RX_N3
A10	GND	B10	GND
A11	GND	B11	TX_P3
A12	GND	B12	TX_N3
A13	GND	B13	GND
A14	+5V	B14	+5V
A15	/RST2	B15	/RST3
A16	SHTTL_PLG2	B16	SHTTL_PLG3
A17	SH_PWR_EN2	B17	SH_PWR_EN3
A18	OCD2	B18	OCD3
A19	/ACT_LED2	B19	/ACT_LED3
A20	/RDY_LED2	B20	/RDY_LED3

Table 23: Pin Assignment of the 40-pin Connector

Signal	Direction	Function
TX_P	Out	Transmit Data positive
TX_N	Out	Transmit Data negative
RX_P	In	Receive Data positive
RX_N	In	Receive Data negative
SH_PWR_Enx	Out	Shuttle Power enable
OCDx	In	Over current detect
/ACT_LEDx	Out	Channel activity
/RDY_LEDx	Out	Ready for take out
+5V	Power	Power supply +3.3V
GND	Ground	Ground

Table 24: Signal Mnemonics of 20-pin Plug Connector

9.1.5 Connector for connecting shuttle P26 and P27

The RAIDStor supports the following pin assignment of the female 20-pin plug connector for connecting the 6210SH or 6211SH to the RAIDStor:

A10	GND	B10	GND
A9	GND	B9	RX_P
A8	GND	B8	RX_N
A7	GND	B7	GND
A6	GND	B6	TX_N
A5	GND	B5	TX_P
A4	/RST	B4	GND
A3	/ACT_LED	B3	/RDY_LED
A2	GND	B2	GND
A1	+5V	B1	+5V

Table 25: Supported Pins of 20-pin Plug Pin Receptacle Connector on Shuttle

NOTE: pinning of connector on shuttle is different because of different connector types.

Signal	Direction	Function
TX_P	Out	Transmit Data positive
TX_N	Out	Transmit Data negative
RX_P	In	Receive Data positive
RX_N	In	Receive Data negative
+5V	Power	Power supply +5V
GND	Ground	Ground

Table 26: Signal Mnemonics of 20-pin Plug Connector

9.1.6 USB Connector J7

A1	VCC
A2	USBP-
A3	USBP+
A4	GND

Table 27: Pin Assignment of the USB Type A Connector

9.1.7 COM1 Connector P7

COM1 is a standard RS232 interface accessible via an RJ45 connector at the front panel. Connector types:

- Modular 8/8-pin mounting jack according to FCC68
- Mating connector: Modular 8/8-pin plug according to FCC68

1	DSR
2	DCD
3	DTR
4	GND
5	RXD
6	TXD
7	CTS
8	RTS

Table 28: Pin Assignment of 8-pin RJ45 RS232 Connector (COM1)

9.1.8 Connector J1

The RAIDStor board supports the following pin assignment of the 110-pin CompactPCI J1 connector. Please refer to the PICMG 2.16 specification for signal mnemonics of the J1 connector. PCI bus signals not used on board but will be pre-charged.

	A	B	C	D	E	F
25	5V	-	-	3.3V	5V	GND
24		5V	-		-	GND
23	3.3V	-	-	5V	-	GND
22		GND	3.3V		-	GND
21	3.3V	-	-		-	GND
20		GND	-		-	GND
19	3.3V	-	-	GND	-	GND
18		GND	3.3V		-	GND
17	3.3V	IPMB_SCL	IPMB_SDA	GND	-	GND
16		GND	-		-	GND
15	3.3V	-	-	/BD_SEL	-	GND
14	Key Area					
13						
12						
11	-	-	-	GND	-	GND
10	-	GND	3.3V	-	-	GND
9	-	-	-	GND	-	GND
8	-	GND	-	-	-	GND
7	-	-	-	GND	-	GND
6	-	-	3.3V	-	-	GND
5	-	-	/PCI_RST	GND	-	GND
4	IPMB_ PWR	-	-	-	-	GND
3	-	-	-	5V	-	GND
2	-	5V		-	-	GND
1	5V	-	-	-	5V	GND

Table 29: Pin Assignment of CompactPCI J1 (110-pin type "B" modified)

9.1.9 Connector J2

The RAIDStor board supports the following pin assignment of the 110-pin CompactPCI J2 connector:

	A	B	C	D	E	F
22	GA4	GA3	GA2	GA1	GA0	GND
21	-	-	-	-	-	GND
20	-	-	-	-	-	GND
19	-	-	-	-	-	GND
18	-	-	-	-	-	GND
17	-	-	/PRST	-	-	GND
16	-	-	-	-	-	GND
15	-	-	-	-	-	GND
14	-	-	-	-	-	GND
13	-	-	-	-	-	GND
12	-	-	-	-	-	GND
11	-	-	-	-	-	GND
10	-	-	-	-	-	GND
9	-	-	-	-	-	GND
8	-	-	-	-	-	GND
7	-	-	-	-	-	GND
6	-	-	-	-	-	GND
5	-	-	-	-	-	GND
4	-	-	-	-	-	GND
3	-	-	-	-	-	GND
2	-	-	-	-	-	GND
1	-	-	-	-	-	GND

Table 30: Pin Assignment of CompactPCI J2 (110-pin type "B" modified)

Signal	Direction	Function
GA4.. GA0	Out	Geographical addresses
/PRST	In	Push button reset

Table 31: Signal Mnemonics of CompactPCI J2

9.1.10 Connector J3

The RAIDStor board supports the following pin assignment of the 95-pin CompactPCI J3 connector:

	A	B	C	D	E	F
19	AC_Path	AC_Path	AC_Path	AC_Path	AC_Path	GND
18	LPa_DA+	LPa_DA-	GND	LPa_DC+	LPa_DC-	GND
17	LPa_DB+	LPa_DB-	GND	LPa_DD+	LPa_DD-	GND
16	LPb_DA+	LPb_DA-	GND	LPb_DC+	LPb_DC-	GND
15	LPb_DB+	LPb_DB-	GND	LPb_DD+	LPb_DD-	GND
14	AC_Path	AC_Path	AC_Path	AC_Path	AC_Path	GND
13	-	-	-	-	-	GND
12	-	-	-	-	-	GND
11	-	-	-	-	-	GND
10	-	-	-	-	-	GND
9	-	-	-	-	-	GND
8	-	-	-	-	-	GND
7	-	-	-	-	-	GND
6	-	-	-	-	-	GND
5	-	-	-	-	-	GND
4	-	-	-	-	-	GND
3	-	-	-	-	-	GND
2	-	-	-	-	-	GND
1	-	-	-	-	-	GND

Table 32: Pin Assignment of CompactPCI J3 (95-pin type "B" modified)

Signal	Direction	Function
LPa_DA±	In/out	Link Port a data pair A
LPa_DB±	In/out	Link Port a data pair B
LPa_DC±	In/out	Link Port a data pair C
LPa_DD±	In/out	Link Port a data pair D
LPb_DA±	In/out	Link Port a data pair A
LPb_DB±	In/out	Link Port a data pair B
LPb_DC±	In/out	Link Port a data pair C
LPb_DD±	In/out	Link Port a data pair D
AC_Path	In/out	Connected with Bypass capacitor to GND

Table 33: Signal Mnemonics of CompactPCI J3

9.1.11 Connector J5

The RAIDStor board supports the following 110-pin CompactPCI J5 connector:

	A	B	C	D	E	F
22	/RESET	GND	USBP2+	USBP1+	USB2+5V	GND
21	USB2GND	USBP2-	USBP1-	USB1+5V	USB1GND	GND
20	-	-	-	-	-	GND
19	GND	-	TxD4	RxD4	-	GND
18	-	-	-	-	-	GND
17	GND	-	TxD3	RxD3	-	GND
16	-	/RI2	/CTS2	/RTS2	/DSR2	GND
15	GND	/DTR2	TxD2	RxD2	/DCD2	GND
14	-	/RI1	/CTS1	/RTS1	/DSR1	GND
13	GND	/DTR1	TxD1	RxD1	/DCD1	GND
12	-	-	-	-	-	GND
11	LILED/ Link_UP A	ACTLED/ Activity A	LILED/ Link_UP B	ACTLED/ Activity B-	-	GND
10	-	-	-	-	-	GND
9	-	-	-	-	-	GND
8	-	-	-	-	-	GND
7	-	-	-	-	-	GND
6	-	-	-	-	-	GND
5	-	-	-	-	-	GND
4	-	-	-	-	-	GND
3	-	-	-	-	-	GND
2	-	-	-	-	-	GND
1	-	-	-	-	-	GND

Table 34: Pin Assignment of CompactPCI J5 (110-pin type "B" modified)

Signal	Direction	Function
USBPx±	In/Out	USB ports
/RESET	In	Reset input
CTSx	In	Clear To Send
DCDx	In	Data Carrier Detected
DSRx	In	Data Set Ready
DTRx	Out	Data Terminal Ready
GND	-	Reference potential
RIx	In	Ring Indicator
RTSx	Out	Request To Send
RXDx	In	Receive data
TXDx	Out	Transmit data

Table 35: Signal Mnemonics of CompactPCI J5

9.2 Shuttle 6210SH & 6211SH

9.2.1 6210SH PATA Connector J1

The 6210SH supports the following pin assignment of the female 44-pin plug connector for connecting one Parallel ATA hard disc drive:

1	/IDE_RST	2	GND
3	IDE_D7	4	IDE_D8
5	IDE_D6	6	IDE_D9
7	IDE_D5	8	IDE_D10
9	IDE_D4	10	IDE_D11
11	IDE_D3	12	IDE_D12
13	IDE_D2	14	IDE_D13
15	IDE_D1	16	IDE_D14
17	IDE_D0	18	IDE_D15
19	GND	20	-
21	IDE_DMARQ	22	GND
23	/IDE_WR	24	GND
25	/IDE_RD	26	GND
27	/IDE_RDY	28	IDE_CS
29	/IDE_DMACK	30	GND
31	IDE_IRQ	32	-
33	IDE_A1	34	-
35	IDE_A0	36	IDE_A2
37	/IDE_CS0	38	/IDE_CS1
39	-	40	GND
41	+5V	42	+5V
43	GND	44	-

Table 36: Supported pins of the Parallel ATA 44-pin plug connector on Shuttle

Signal	Direction	Function
/IDE_RST	Out	Reset
IDE_D15.. IDE_D0	In/Out	Data
IDE_DMARQ	In	DMA request
/IDE_DMACK	Out	DMA acknowledge
/IDE_WR	Out	IO write signal
/IDE_RD	Out	IO read signal
/IDE_RDY	In	IO ready
IDE_A2.. IDE_A0	Out	Addresses
/IDE_CS1.. /IDE_CS0	Out	Chip selects
IDE_IRQ	In	Interrupt request
GND		Ground

Table 37: Signal Mnemonics of the Parallel ATA 44-pin Plug Connector on Shuttle

9.2.2 6211SH SATA Connector J1

The 6211SH supports the following pin assignment of the female 22-pin plug connector for connecting one Serial ATA hard disc drive:

1	GND
2	HOST_RX_P
3	HOST_RX_N
4	GND
5	HOST_TX_N
6	HOST_TX_P
7	GND
8	NC
9	NC
10	NC
11	GND
12	GND
13	GND
14	VCC (5V)
15	VCC (5V)
16	VCC (5V)
17	GND
18	GND
19	GND
20	NC
21	NC
22	NC

Table 38: Supported pins of Serial ATA 22-pin plug connector on Shuttle

Signal	Direction	Function
HOST_TX_P	In	Host Transmit Data positive
HOST_TX_N	In	Host Transmit Data negative
HOST_RX_P	Out	Host Receive Data positive
HOST_RX_N	Out	Host Receive Data negative
VCC	Power	Power supply (+5V)
GND	Ground	Ground
NC	None	Not Connected

Table 39: Signal Mnemonics of the Serial ATA 22-pin Plug Connector on Shuttle

9.2.3 6210SH and 6211SH Connector J2

The 6210SH and 6211SH support the following pin assignment of the female 20-pin plug connector for connecting the shuttle to the RAIDStor:

A1	GND	B1	GND
A2	GND	B2	TX_P
A3	GND	B3	TX_N
A4	GND	B4	GND
A5	GND	B5	RX_N
A6	GND	B6	RX_P
A7	/RST	B7	GND
A8	/ACT_LED	B8	/RDY_LED
A9	GND	B9	GND
A10	+5V	B10	+5V

Table 40: Supported Pins of 20-pin Plug Pin Receptacle Connector on Shuttle

NOTES:

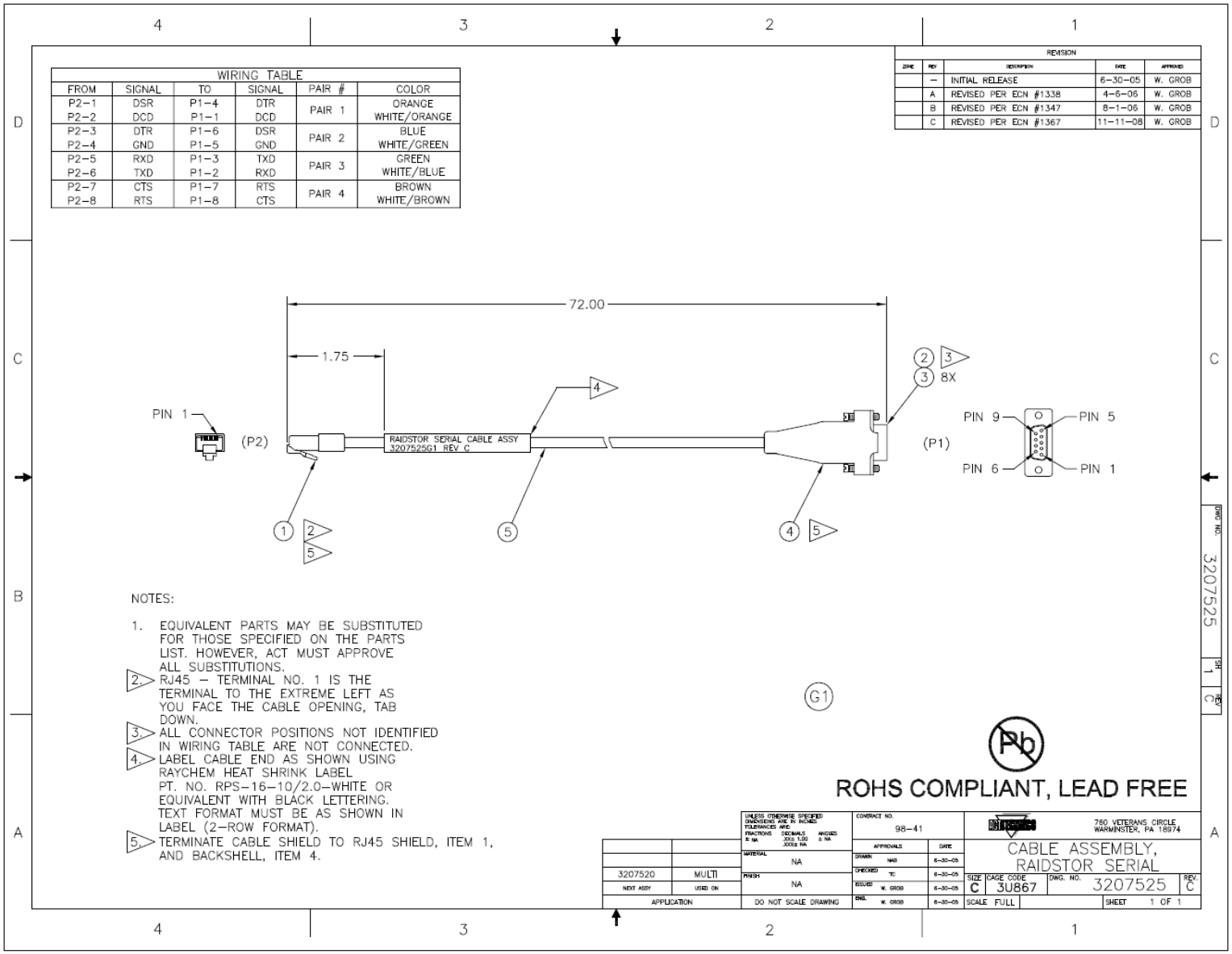
1. Pinning of connectors for the shuttle and the RAIDStor baseboard are different because of different types of connectors.

2. +3.3V may be generated on shuttle if necessary!

Signal	Direction	Function
TX_P	Out	Transmit Data positive
TX_N	Out	Transmit Data negative
RX_P	In	Receive Data positive
RX_N	In	Receive Data negative
/RST	In	Reset active low
+5V	Power	Power supply +5V
/ACT_LED	In	LED for drive activity
/RDY_LED	In	LED for shuttle removal ready status
GND	Ground	Ground

Table 41: Signal Mnemonics of 20-pin Plug Connector

9.3 Console Cable



10 Memory Map

The memory map of the board complies with the PowerPC CHRP (Common Hardware Reference Platform) Specification. The integrated host-to-PCI bridge is set to map B to support this mapping.

CPU Address Range	Size	Description
0x0000_0000..1FFF_FFFF	512MB	DRAM
0x2000_0000..6FFF_FFFF	1.25GB	Reserved
0x7000_0000..7FFF_FFFF	256MB	CPLD(ROM-X) Interface
0x8000_0000..FCFF_FFFF	2GB-48MB	PCI Memory Space
0xFD00_0000..FDFF_FFFF	16MB	PCI ISA Memory Space
0xFE00_0000..FE00_FFFF	64KB	PCI ISA I/O Space
0xFE80_0000..FEBF_FFFF	4MB	PCI I/O Space (not used)
0xFEC0_0000..FEDF_FFFF	2MB	PCI Config Addr. Reg.
0xFEE0_0000..FEEF_FFFF	1MB	PCI Config Data Reg.
0xFEF0_0000..FEFF_FFFF	1MB	PCI IACK Space
0xFFE0_0000..FFFF_FFFF	2MB	Boot Flash (8-bit)

Table 42: CPU Address Map

10.1 PCI/ISA I/O Space Memory Map

This memory map complies with the ISA I/O address assignments. Refer to data sheet "ALADDIN M1543: Desktop South Bridge, version 1.25, Jan. 1998" for configuration registers. The Ali1543 is mapped through the PCI Bridge TI2050 on a second bus (not PCI bus zero of the PPC).

CPU Address Range	Device	Register
0xFE00_0000..FE00_000F	M1543	DMA1 (slave)
0xFE00_0020	M1543	INT_1 (master) Control Register
0xFE00_0021	M1543	INT_1 (master) Mask Register
0xFE00_0040	M1543	Timer Counter - Channel 0 Count
0xFE00_0041	M1543	Timer Counter - Channel 1 Count
0xFE00_0042	M1543	Timer Counter - Channel 2 Count
0xFE00_0043	M1543	Timer Counter Command Mode Register
0xFE00_0060	M1543	Read_access Clear IRQ[12] (for PS2),
0xFE00_0060	M1543	IRQ[1] Latched Status

CPU Address Range	Device	Register
0xFE00_0060	M1543	Keyboard Data Buffer
0xFE00_0061	M1543	NMI and Speaker Status and Control
0xFE00_0064	M1543	Keyboard Status(R)/Command(W)
0xFE00_0080..FE00_009F	M1543	DMA Channel x Page Register
0xFE00_00A0	M1543	INT_2 (slave) Control Register
0xFE00_00A1	M1543	INT_2 (slave) Mask Register
0xFE00_00C0..FE00_00DF	M1543	DMA2 (master)
0xFE00_00F0	M1543	Coprocessor Error Ignored Register
0xFE00_0170..FE00_0177	M1543	IDE Secondary registers part A
0xFE00_01F0..FE00_01F7	M1543	IDE Primary registers part A
0xFE00_02F8..FE00_02FF	M1543 Super I/O	UART2 Controller
0xFE00_0378..FE00_037F	M1543 Super I/O	Parallel Port Controller
0xFE00_03F0	M1543 Super I/O	Config Port Index
0xFE00_03F1	M1543 Super I/O	Config Port Data
0xFE00_0376..FE00_0377	M1543	IDE Secondary registers part B
0xFE00_03F6..FE00_03F7	M1543	IDE Primary registers part B
0xFE00_03F8..FE00_03FF	M1543 Super I/O	UART1 Controller
0xFE00_040B	M1543	DMA1 Extended Mode Register
0xFE00_0481..FE00_048B	M1543	DMA High Page Registers
0xFE00_04D0	M1543	INT_1 (master) Edge/Level Control
0xFE00_04D1	M1543	INT_2 (slave) Edge/Level Control
0xFE00_04D6	M1543	DMA2 Extended Mode Register
0xFE00_1800..FE00_181E	M1543	SMB Controller
0xFE00_2000..FE00_201F	M1543	PMU of ALI
0xFE00_F000..FE00_F00F	M1543	IDE bus master registers

Table 43: PCI I/O Address Map

11 ACT/Technico MIB(s)

11.1 MIB Conventions

The RAIDStor MIB will utilize the following variables throughout the MIB:

- Operational status (Up, Down) – Read only
- Administrative state (Up, Down) – Read write.
- Row status

Operational Status is a read-only object that reports the operating status of the entity in question. This object is only used to query status – it is not used for control.

Administrative state is a read-write object used for controlling the behavior of a piece of hardware. An AdminState of Up is equivalent to Active and an AdminState of Down is equivalent to standby.

Each MIB table may have multiple Rows, each corresponding to an entity in the system. For example, the RAIDStor table will have two rows if two RAIDStor blades are present. For each pair of redundant devices, only one of the pair can have an administrative status of Up at a time – this is the active entity.

Changing the **AdminState** on the primary device from Up to Down will force a fail-over to the standby device. If no standby device is available (present and **OperStatus** = Up), this will cause the system to stop functioning.

Changing the **AdminStat**e on the secondary device from Down to Up is not allowed. The standby device is not authorized to initiate a failover.

The **AdminState** value of testing is used to force a device off-line prior to running diagnostics tests on the device.

RowStatus allows the manager to add and remove items from the MIB by creating and deleting rows from the Tables in the MIB. This variable will be used in the trap configuration table. This may be the only table in which it is used for the first release of the SNMP management system

11.2 ACT/Technico Corporate MIB

```
--****************************************************************
-- MIB ACTTECHNICO-CORP-MIB from project `raidstore'
--
-- Auto-generated by SNMP IQ Application Builder on 03/12/2004
-- Version:
-- Authored on: 03/11/2004
--
-- The root of the ACTTECHNICO Systems enterprise MIB.
--
--
--****************************************************************
--

ACTTECHNICO-CORP-MIB DEFINITIONS ::= BEGIN

    IMPORTS
        --============================================================
        --
        --     Imports from: SNMPv2-SMI
        --
        --============================================================
        MODULE-IDENTITY,
        enterprises FROM SNMPv2-SMI;


--============================================================
```

```
acttechnicoCorp      MODULE-IDENTITY
    LAST-UPDATED "200605090000Z"
    ORGANIZATION "ACT Technico"
    CONTACT-INFO
        "Ken Grob
         ACT/Technico
         760 Veterans Circle
         Warminster, PA 18974
         215.956.1200"
    DESCRIPTION
        "The root of the ACTTECHNCIO Systems enterprise MIB."
    REVISION "200605090000Z"
    DESCRIPTION
        "First customer version."
    ::= { enterprises 20036 }



-- Top-level structure of the MIB

acttechnicoCorpAdmin            OBJECT IDENTIFIER ::= { acttechnicoCorp 1 }
acttechnicoProjectOID           OBJECT IDENTIFIER ::= { acttechnicoCorpAdmin 1 }
acttechnicoProductOID           OBJECT IDENTIFIER ::= { acttechnicoCorpAdmin 2 }
acttechnicoMibModules           OBJECT IDENTIFIER ::= { acttechnicoCorpAdmin 6 }
acttechnicoProjects             OBJECT IDENTIFIER ::= { acttechnicoCorp 2 }
acttechnicoProducts             OBJECT IDENTIFIER ::= { acttechnicoCorp 3 }
acttechnicoExperimental         OBJECT IDENTIFIER ::= { acttechnicoCorp 4 }

END
```

11.3 ACT/Technico RAIDStor MIB

```
--*****************************************************************
-- MIB acttechnicoRAIDStor
--
-- Auto-generated by SNMP IQ Application Builder on 09/29/2004
-- Version:
-- Authored on: 03/11/2004
--
-- ACT Technico RAIDStor Systems Definition
--
--*****************************************************************
--

ACTTECHNICO-RAIDStor-MIB DEFINITIONS ::= BEGIN

    IMPORTS
        --============================================================
        --
        --     Imports from: ACTTECHNICO-CORP-MIB
        --
        --============================================================
        acttechnicoProjectOID FROM ACTTECHNICO-CORP-MIB

        --============================================================
        --
        --     Imports from: SNMPv2-SMI
```

```
        --
        --================================================================
        Counter32,
        Counter64,
        Gauge32,
        Integer32,
        IpAddress,
        MODULE-IDENTITY,
        NOTIFICATION-TYPE,
        OBJECT-IDENTITY,
        OBJECT-TYPE,
        Opaque,
        TimeTicks,
        Unsigned32 FROM SNMPv2-SMI

        --================================================================
        --
        --     Imports from: SNMPv2-TC
        --
        --================================================================
        DateAndTime,
        DisplayString,
        RowStatus,
        TEXTUAL-CONVENTION,
        TruthValue FROM SNMPv2-TC;


--================================================================
acttechnicoRAIDStor MODULE-IDENTITY
    LAST-UPDATED "200607110000Z"
    ORGANIZATION "ACT Technico"
    CONTACT-INFO
        "support@acttechnico.com"
    DESCRIPTION
        "ACT Technico RAIDStor Systems Definition"
    REVISION "200607110000Z"
    DESCRIPTION
        "Single Blade Release, use agent V1.2 or higher."
    REVISION "200606280000Z"
    DESCRIPTION
        "Single Blade Mods, use agent V1.1 or higher."
    REVISION "200605310000Z"
    DESCRIPTION
        "Initial customer version."
    ::= { acttechnicoProjectOID 1 }

-- ************************************************************************
-- TEXTUAL CONVENTIONS
-- ************************************************************************

RsDiskID ::= TEXTUAL-CONVENTION
    STATUS       current
    DESCRIPTION
        "ID of a disk drive on a RAIDStor blade."
    SYNTAX   INTEGER {
        diskA(1),
        diskB(2),
        diskC(3),
        diskD(4),
```

```
        diskNA(5) }

RsOAStatus ::= TEXTUAL-CONVENTION
    STATUS       current
    DESCRIPTION
        "
         Operational or Administrative status of an object in the RAIDStor MIB.
        "
    SYNTAX   INTEGER {
        rsUP(1),
        rsDown(2),
        rsTesting(3) }

RsTrapType ::= TEXTUAL-CONVENTION
    STATUS       current
    DESCRIPTION
        "Types of traps supported in the RAIDStor MIB."
    SYNTAX   INTEGER {
        rsNetworkSyncTrap(1),
        rsDiskSyncTrap(2),
        rsRAIDStoreStateChangeTrap(3),
        rsNASStateChangeTrap(4),
        rsNICUpTrap(5),
        rsNICDownTrap(6),
        rsDiskStatusChangeTrap(7) }


-- Top-level structure of the MIB

rsEvent                    OBJECT-IDENTITY
    STATUS    current
    DESCRIPTION
        "RAIDStor Events"
    ::= { acttechnicoRAIDStor 0 }

rsTrapNetworkSync       NOTIFICATION-TYPE
    OBJECTS {
       rsLastTrapSent,
       rsAlarmTimeStamp,
       rsNASStartSync,
       rsNASSyncStatus,
       rsNASSyncPercent
    }
    STATUS current
    DESCRIPTION
        "
         Network Sync event. Details of the sync status can be read
         from the rsNASConfigTable. A network sync will only occur
         on a dual-blade RAIDStor.
        "
    ::= { rsEvent 1 }

rsTrapDiskSync          NOTIFICATION-TYPE
    OBJECTS {
       rsLastTrapSent,
       rsAlarmTimeStamp,
       rsStartSync,
       rsSyncStatus,
       rsSyncPercent
```

```
    }
    STATUS current
    DESCRIPTION
        "
         Local Disk Sync event.  A local disk sync will occur when a
         RAIDStor rebuilds a local disk drive.
        "
    ::= { rsEvent 2 }

rsTrapRAIDStorStateChg  NOTIFICATION-TYPE
    OBJECTS {
       rsLastTrapSent,
       rsAlarmTimeStamp,
       rsOpStatus,
       rsAdminState
    }
    STATUS current
    DESCRIPTION
        "
         RAIDStor State Change. This trap will occur when
         rsStatus.0 changes. This indicates that heartbeat
         and/or drbd have stopped, or both have started.
        "
    ::= { rsEvent 3 }

rsTrapNASStateChange    NOTIFICATION-TYPE
    OBJECTS {
       rsLastTrapSent,
       rsAlarmTimeStamp,
       rsNASOpStatus,
       rsNASAdminState
    }
    STATUS current
    DESCRIPTION
        "
         NAS State Change. Details of the NAS state can be read from
         the rsNASConfigTable
        "
    ::= { rsEvent 4 }

rsTrapNICUp             NOTIFICATION-TYPE
    OBJECTS {
       rsLastTrapSent,
       rsAlarmTimeStamp,
       rsMacAddress
    }
    STATUS current
    DESCRIPTION
        "Network Interface Up trap"
    ::= { rsEvent 5 }

rsTrapNICDown           NOTIFICATION-TYPE
    OBJECTS {
       rsLastTrapSent,
       rsAlarmTimeStamp,
       rsMacAddress
    }
    STATUS current
    DESCRIPTION
```

```
        "Network Interface Down trap"
    ::= { rsEvent 6 }

rsTrapDiskStatusChange         NOTIFICATION-TYPE
    OBJECTS {
       rsLastTrapSent,
       rsAlarmTimeStamp,
       rsHDDOpSts
    }
    STATUS current
    DESCRIPTION
        "
         DiskStatusChangeTrap. This trap will occur when a disk
         is taken off line or put on-line. This will normally be
         when a disk shuttle is removed and/or replaced.
        "
    ::= { rsEvent 7 }

rsHardware                 OBJECT-IDENTITY
    STATUS    current
    DESCRIPTION
        "System Info"
    ::= { acttechnicoRAIDStor 1 }

rsBladeInfo                OBJECT-IDENTITY
    STATUS    current
    DESCRIPTION
        "Blade Information"
    ::= { rsHardware 1 }


rsIpAddress     OBJECT-TYPE
    SYNTAX      IpAddress
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Ip Address of the local blade interface"

    ::= { rsBladeInfo 1 }


rsOpStatus      OBJECT-TYPE
    SYNTAX      RsOAStatus
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "
         Operational Status of the RAIDStor blade:

         On a dual-blade RAIDStor rsUP is returned
         if both DRBD and Heartbeat are running,
         else rsDown is returned.

         On a single blade RAIDStor rsUP is always
         returned.
        "

    ::= { rsBladeInfo 2 }
```

```
rsAdminState    OBJECT-TYPE
    SYNTAX      RsOAStatus
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "
         Administrative State of the RAIDStor blade:

           This entry can be used to manually stop
         or start heartbeat.
        "

    ::= { rsBladeInfo 3 }


rsSerialNumber  OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "
         Serial Number of the RAIDStor blade, this is read from
         the EEPROM, it is also available in the /proc filesystem.
        "

    ::= { rsBladeInfo 4 }


rsStartSync     OBJECT-TYPE
    SYNTAX      INTEGER {
                    rsNoSynch(1),
                    rsQuickSync(2),
                    rsFullSynch(3)
                    }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "
         Start a synchronization from the primary to the secondary HDD.
           This feature is no longer used and should be ignored.
        "
    ::= { rsBladeInfo 5 }


rsSyncStatus    OBJECT-TYPE
    SYNTAX      INTEGER {
                    rsSyncStatusNone(1),
                    rsSyncStatusRecv(2),
                    rsSyncStsXmit(3)
                    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "
         SYNC Status. A retuned valuse of rsSyncStsXmit(3) indicates
         that the system is actively rebuilding a local disk drive.
        "
```

```
    ::= { rsBladeInfo 6 }


rsSyncPercent   OBJECT-TYPE
    SYNTAX      Integer32 (0..100)
    UNITS       "Percent"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "
         Sync Completion Percentage. This field indicates an
         approximation of the amount of the partition that has
         been re-built.
        "

    DEFVAL { 0 }
    ::= { rsBladeInfo 7 }


rsLastFailOver  OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "
         On a dual-blade RAIDStor the last 'failover' is displayed.
         A failover is defined as RAIDStor blade tading control
         of the public NAS Partition. When the failed primary blade
         is returned to service the RAIDStor will fail back to the
         primary. The failure of the backup blade when it is not
         actively serving the public NAS will not cause a failover,
         but it will generate a trap.
        "

    ::= { rsBladeInfo 8 }


rsSoftwareVersion OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "
         Software Version of the RAIDStor blade, this is read from
         the raidstor.cfg file that is used for persistant storage
         for the RAIDStor. The software version included the date
         and the revision control number for this release.
        "

    ::= { rsBladeInfo 9 }


rsHardwareVersion OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "
         Hardware Version of the RAIDStor blade, this is
```

```
         read from data loaded into the RAIDStor during
         board assembly. This information is also available
         in the /proc filesystem.
        "

    ::= { rsBladeInfo 10 }


rsUptime        OBJECT-TYPE
    SYNTAX      TimeTicks
    UNITS       "1/100th of a second"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "
         Time, in 1/100ths of a second since the snmp agent
         was last restarted. Standard unit
         for Time ticks is 1/100th of a second.
        "

    ::= { rsBladeInfo 11 }


rsMacAddress    OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(25))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "
         MAC Address for the RAIDStor blade. There are two
         Ethernet ports on the RAIDStor, but due to the bonding
         driver only one MAC address will be seen.
        "

    ::= { rsBladeInfo 12 }


rsNumDrives     OBJECT-TYPE
    SYNTAX      Integer32 (1..4)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "
         Number of HDDs on the RAIDStor blade. Up to 2 on the baseboard
         and two additional on the expansion board.
        "


    ::= { rsBladeInfo 13 }


rsHostname      OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "
         Blade name, this is normally primary or backup. This should
         not be changed
```

```
        "

    ::= { rsBladeInfo 14 }


rsPartnerTimeout OBJECT-TYPE
    SYNTAX      Integer32 (0..120)
    UNITS       "Seconds"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "
         How long a RAIDStor blade will wait for its partner before assuming
         the partner will not arrive. Units is number of seconds. A value of
         0 indicates the blade should wait forever. This only applies to
         dual blade RAIDStors.
        "

    ::= { rsBladeInfo 15 }


rsRaidstorReset OBJECT-TYPE
    SYNTAX      Integer32 (0..2)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "
         Reset the RAIDStor card.
         Write a 1 for a reboot.
         Write a 2 to halt the RAIDStor.
          "

    ::= { rsBladeInfo 16 }


rsPartNumber    OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "
         Part number of the RAIDStor blade. The full part number
         for the RAIDStor is not currently loaded at the factory.
        "

    ::= { rsBladeInfo 17 }

rsDiskInfo                OBJECT-IDENTITY
    STATUS    current
    DESCRIPTION
        "Disk drive info"
    ::= { rsHardware 2 }


-- Begin Table rsDiskTable
==================================================
rsDiskTable     OBJECT-TYPE
    SYNTAX      SEQUENCE OF DiskEntry
    MAX-ACCESS  not-accessible
```

```
    STATUS      current
    DESCRIPTION
        "The Disk table contains information on the disk drives"

    ::= { rsDiskInfo 1 }


diskEntry       OBJECT-TYPE
    SYNTAX      DiskEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Row of table diskTable"

    INDEX {
         rsHDDNum
        }
    ::= { rsDiskTable 1 }


DiskEntry               ::= SEQUENCE {
        rsHDDNum                                Integer32,
        rsHDDOpSts                              RsOAStatus,
        rsHDDAdminState                         RsOAStatus,
        rsCapacity                              Integer32,
        rsHDDDeviceName                         DisplayString
    }

rsHDDNum        OBJECT-TYPE
    SYNTAX      Integer32 (1..4)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "ID number of the hard disk on the RAIDStor blade"

    ::= { diskEntry 1 }

rsHDDOpSts      OBJECT-TYPE
    SYNTAX      RsOAStatus
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Hard drive Operational Status"

    ::= { diskEntry 2 }


rsHDDAdminState OBJECT-TYPE
    SYNTAX      RsOAStatus
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Hard Drive admin State"

    ::= { diskEntry 3 }

rsCapacity      OBJECT-TYPE
    SYNTAX      Integer32 (1..5000000)
    UNITS       "Megabytes"
```

```
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Total capacity in Megabytes of this hard drive"

    ::= { diskEntry 4 }


rsHDDDeviceName OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The Device Name for this hard drive. For example:
        /dev/ide/proc/hda"

    ::= { diskEntry 5 }


-- End Table rsDiskTable
=====================================================

rsTemperature   OBJECT-TYPE
    SYNTAX      Integer32 (-300..1300)
    UNITS       "Degrees C"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Temperature of the RAIDStorTemperature in degrees centigrade (C)."

    ::= { rsHardware 4 }

rsNASConfiguration         OBJECT-IDENTITY
    STATUS    current
    DESCRIPTION
        "Network Attached Storage Information"
    ::= { acttechnicoRAIDStor 2 }


rsNetworkVirtualIPAddress OBJECT-TYPE
    SYNTAX      IpAddress
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "
         Virtual IP address of the RAIDStor NAS, on a dual blade
         RAIDStor this only exists on one blade at any given time.
         On a single blade RAIDStor the virtual address is the same
         as the local bonding driver address.

         Single Blade returns the address of bond0
         Dual blade returns the address of bond0:0
        "

    ::= { rsNASConfiguration 2 }


rsNASOpStatus   OBJECT-TYPE
    SYNTAX      RsOAStatus
```

```
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Operational Status of the RAIDStor NAS"

    ::= { rsNASConfiguration 3 }


rsNASAdminState OBJECT-TYPE
    SYNTAX      RsOAStatus
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Admin State of the RAIDStor NAS"

    ::= { rsNASConfiguration 4 }


rsNASRaidType   OBJECT-TYPE
    SYNTAX      INTEGER {
                    rsRAID1(1),
                    rsRAIDDisable(2)
                    }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "RAID Type for the RAIDStor NAS"

    ::= { rsNASConfiguration 5 }


rsPartnerIP     OBJECT-TYPE
    SYNTAX      IpAddress
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "
         RAIDStor Blade IP Address of other blade, on a single
         blade this will return 0.0.0.0.
        "

    ::= { rsNASConfiguration 6 }


rsNASStartSync  OBJECT-TYPE
    SYNTAX      INTEGER {
                    rsNASNoSync(1),
                    rsNASQuickSync(2),
                    rsNASFullSync(3)
                    }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "
         For a synchronization from the primary to the secondary NAS.

         1 = no synchronization
         3 = start a full synchronization
```

```
         You don't really want to do this except for testing as this
         will take a very long time and use up valuable network
         bandwidth.
       "

    ::= { rsNASConfiguration 7 }


rsNASSyncStatus OBJECT-TYPE
    SYNTAX      INTEGER {
                    rsNASSyncStsNone(1),
                    rsNASSyncStsRcv(2),
                    rsNASSyncStsXmit(3)
                    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Sync Status"

    ::= { rsNASConfiguration 8 }


rsNASSyncPercent OBJECT-TYPE
    SYNTAX      Integer32 (0..100)
    UNITS       "Percent"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "
         Sync completion percentage for the RAIDStor NAS, this is
         an estimate that is updated about every 30 seconds.
        "

    ::= { rsNASConfiguration 9 }


rsNASLastFailOver OBJECT-TYPE
    SYNTAX      DateAndTime
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Time Stamp of the last fail over"

    ::= { rsNASConfiguration 10 }


rsNASNetRaidProtocol OBJECT-TYPE
    SYNTAX      INTEGER {
                    rsNASDrbdProtocolA(1),
                    rsNASDrbdProtocolB(2),
                    rsNASDrbdProtocolC(3)
                    }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "
        Dual blade only
        Controls the operations of Net RAIDwrite operations
```

```
        Protocol A: A write operation is complete as soon as the data is written
        to disk and sent to the network.

        Protocol B: A write operation is complete as soon as a reception
        acknowledgement arrives.

        Protocol C: A write operation is complete as soon as a write
        acknowledgement arrives.

        Only 'C' is supported by ACT at this time. If you want something
        else contact ACT/Technico engineering.
        "

    ::= { rsNASConfiguration 11 }


rsNASSystemVersion OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "
         This only applies to dual blade RAIDStor.
         Version of the NAS system. This is the version number for the system as
         a whole and is distinct from the version number of the RAIDStor blade.
        "

    ::= { rsNASConfiguration 12 }


rsNASSystemHealthSummary OBJECT-TYPE
    SYNTAX      INTEGER {
                    green(1),
                    yellow(2),
                    red(3)
                    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "
         Summary of the health of the NAS system. More details on the status of
         the system are available in the rsStatus section of the MIB
         1 == all drives and Ethernet ports are up
         2 == a redundant device is down, but another spare is available
         3 == a device is down and a second failure in this area will
              render the RAIDStor pair down for dual blade or the RAIDStor
              down for single blade.
         A RAIDStor is defined as down when it can not serve the public IP
         address.
        "

    ::= { rsNASConfiguration 13 }

rsNASSyncMaxBW  OBJECT-TYPE
    SYNTAX      Gauge32
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "
```

```
          Dual blade configuration item only

          Maximum network bandwidth used during synchronization. KB/S
         "

    DEFVAL { 100000 }
    ::= { rsNASConfiguration 15 }


rsNASXferBufferLength OBJECT-TYPE
    SYNTAX      Gauge32
    UNITS       "KB"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "
         Dual blade configuration item only.
         Size in bytes of the TCP transmit buffer in kb
        "

    DEFVAL { 128 }
    ::= { rsNASConfiguration 17 }


rsNASLinkCheckInterval OBJECT-TYPE
    SYNTAX      Gauge32
    UNITS       "sec"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "
         Dual blade only.
         Retry interval for checking the network link. seconds
        "

    DEFVAL { 10 }
    ::= { rsNASConfiguration 18 }

rsSystemInformation        OBJECT-IDENTITY
    STATUS    current
    DESCRIPTION
        "
         RAIDStor system information. The following items
           defined in MIB II system group are system data:
           sysDescr, sysObjectID, sysContact, sysName,
           sysLocation, sysServices
        "
    ::= { acttechnicoRAIDStor 3 }


rsTimeZone      OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Time Zone"

    ::= { rsSystemInformation 1 }
```

```
rsDateAndTime   OBJECT-TYPE
    SYNTAX      DateAndTime
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Current system date and time

        Stored in the form used by the unix date command"

    ::= { rsSystemInformation 2 }

rsStatus                    OBJECT-IDENTITY
    STATUS    current
    DESCRIPTION
        "Status information on various elements in the system"
    ::= { acttechnicoRAIDStor 4 }


rsBladeDiskFreeSpaceRoot   OBJECT-TYPE
    SYNTAX      Integer32
    UNITS   "Megabytes"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "
          This is the free space in the /(root) filesystem. This is actually
          a raid partition that spans multiple drives. The / filesystem
          is normally only 1GB so there is not much free space typically.
          "

    ::= { rsStatus 1 }


rsBladeDiskFreeSpaceUser OBJECT-TYPE
    SYNTAX      Integer32
    UNITS   "Megabytes"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "
          This is the free space in the /raidstor filesystem. This is actually
          a raid partition that spans multiple drives.  On a dual-blade
          RAIDStor this is normally /dev/drbd/1. On a single-blade
          RAIDStor this is /dev/md3.

          The snmp agent code picks this up from a perl script that
          a user could modify to report a filesystem mounted with a
          different name.
          "

    ::= { rsStatus 2 }


rsBladeNetworkConnections OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
```

```
        "
        This is currently hard coded to return 5.
        "

    ::= { rsStatus 3 }

rsBladeIsSingleBlade OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "
          This value is 1 for a Single Blade Raidstor. A RAIDStor
          is considered to be a Single Blade RAIDStor is the run level
          is configured below run level 4 in /etc/initab.

          The SNMP Agent code has some conditional write logic that
          is enabled for a dual-blade RAIDStor. It is not possible to
          write some changes to a dual-blade RAIDStor if the partner
          blade is off line for some reason. The solution to this problem
          is to put the partner blade back on-line.
        "

    ::= { rsStatus 4 }

rsProtocolsAndServices     OBJECT-IDENTITY
    STATUS    current
    DESCRIPTION
        "Information regarding the RAIDStor protocols and services"
    ::= { acttechnicoRAIDStor 5 }

rsNetworkSettings          OBJECT-IDENTITY
    STATUS    current
    DESCRIPTION
        "Network Settings for the RAIDStor NAS"
    ::= { rsProtocolsAndServices 2 }


rsNetworkGateway OBJECT-TYPE
    SYNTAX      IpAddress
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "IP address of the NAS Gateway"

    ::= { rsNetworkSettings 1 }


rsNetworkDNSServer1 OBJECT-TYPE
    SYNTAX      IpAddress
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "IP address of the first DNS Server"

    ::= { rsNetworkSettings 2 }


rsNetworkDNSServer2 OBJECT-TYPE
```

```
    SYNTAX      IpAddress
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "IP address of the first DNS Server"

    ::= { rsNetworkSettings 3 }


rsNetworkDNSServer3 OBJECT-TYPE
    SYNTAX      IpAddress
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "IP address of the third DNS Server"

    ::= { rsNetworkSettings 4 }


rsNetworkHostname OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "
         Host name of the RAIDStor NAS, this is an alternate
         hostname that can be assigned to the public NAS
         partition. This is here because on a dual blade raidstor
         the NAS partition is at a different addres than the blade
         serving the address.
        "


    ::= { rsNetworkSettings 5 }


rsNetworkDomainname OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Domain Name of the RAIDStor NAS"

    ::= { rsNetworkSettings 6 }

rsNotificationConfig       OBJECT-IDENTITY
    STATUS    current
    DESCRIPTION
        "SNMP trap and email configuration"
    ::= { acttechnicoRAIDStor 6 }


-- Begin Table rsTrapConfigTable
=======================================================
rsTrapConfigTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF RsTrapConfigEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
```

```
        "Trap Configuration Table

        This table contains information on the managers that may
        receive traps from this agent. "

    ::= { rsNotificationConfig 1 }


rsTrapConfigEntry OBJECT-TYPE
    SYNTAX      RsTrapConfigEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Row of table rsTrapConfigTable"

    INDEX {
         rsTrapDestinationIndex
        }
    ::= { rsTrapConfigTable 1 }


RsTrapConfigEntry       ::= SEQUENCE {
        rsTrapDestinationIndex                Integer32,
        rsTrapConfigurationAdminState         RsOAStatus,
        rsTrapConfigDestIP                    IpAddress,
        rsTrapConfigRowStatus                 RowStatus
    }

rsTrapDestinationIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..10)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Trap Destination Index

        This table contains a list of manager that traps can be sent to.
        Up to 10 managers may receive traps."

    ::= { rsTrapConfigEntry 1 }


rsTrapConfigurationAdminState OBJECT-TYPE
    SYNTAX      RsOAStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Trap Configuration Admin State"

    ::= { rsTrapConfigEntry 2 }


rsTrapConfigDestIP OBJECT-TYPE
    SYNTAX      IpAddress
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "IP Address of Destination"

    ::= { rsTrapConfigEntry 3 }
```

```
rsTrapConfigRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Row Status for dynamically allocating entries"

    ::= { rsTrapConfigEntry 4 }


-- End Table rsTrapConfigTable
=======================================================

-- Begin Table rsTrapLogTable
=====================================================
rsTrapLogTable  OBJECT-TYPE
    SYNTAX      SEQUENCE OF RsTrapLogEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This table keeps a log of the last 200 traps sent. The index field
        is rsLastTrapSent. Each time a trap is sent the event is stored
        in this table. When the table reaches 200 stored events it
        wraps and begins overwriting the oldest traps. NOTE: The
        index field continues to increment in value even though the
        stored entries wrap at 200."

    ::= { rsNotificationConfig 2 }


rsTrapLogEntry  OBJECT-TYPE
    SYNTAX      RsTrapLogEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Row of table rsTrapLogTable"

    INDEX {
         rsTrapLogTableIndex
        }
    ::= { rsTrapLogTable 1 }


RsTrapLogEntry          ::= SEQUENCE {
        rsTrapLogTableIndex                 Unsigned32,
        rsTrapLogTrapType                   RsTrapType,
        rsTrapLogTimeStamp                  DateAndTime
    }

rsTrapLogTableIndex OBJECT-TYPE
    SYNTAX      Unsigned32 (1..4294967295)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Index of the TrapLogTable.  The index value corresponds to the
         rsLastTrapSent value for each trap."
```

```
    ::= { rsTrapLogEntry 1 }


rsTrapLogTrapType OBJECT-TYPE
    SYNTAX      RsTrapType
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The type of the trap that was sent."

    ::= { rsTrapLogEntry 2 }


rsTrapLogTimeStamp OBJECT-TYPE
    SYNTAX      DateAndTime
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Date and time when the trap was sent"

    ::= { rsTrapLogEntry 3 }


-- End Table rsTrapLogTable
=======================================================

rsAlarmTimeStamp OBJECT-TYPE
    SYNTAX      DateAndTime
    MAX-ACCESS  accessible-for-notify
    STATUS      current
    DESCRIPTION
        "Time Stamp of when the most recent trap was sent"

    ::= { rsNotificationConfig 3 }


rsLastTrapSent  OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "
        Index of the last trap sent by the SNMP agent. Each time a trap is sent
        the SNMP agent increments this value and includes it in the trap being
        sent. An SNMP manager can poll for the field to determine if any traps
        have been missed. This object will continue to increment, but the
        rsTrapLogTable will only contain the last 200 traps, after which point
        the table will wrap.
        "

    ::= { rsNotificationConfig 4 }


-- Begin Table rsEmailNotificationConfigTable
=================================================
rsEmailNotificationConfigTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF RsEmailNotificationConfigEntry
    MAX-ACCESS  not-accessible
    STATUS      current
```

```
    DESCRIPTION
        "Email Notification Configuration Table

        This table contains information on the email addresses that
        traps are sent to. "

    ::= { rsNotificationConfig 5 }


rsEmailNotificationConfigEntry OBJECT-TYPE
    SYNTAX      RsEmailNotificationConfigEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Row of table rsEmailNotificationConfigTable"

    INDEX {
         rsEmailNotificationIndex
        }
    ::= { rsEmailNotificationConfigTable 1 }


RsEmailNotificationConfigEntry ::= SEQUENCE {
        rsEmailNotificationIndex               Integer32,
        rsEmailNotificationAdminState          RsOAStatus,
        rsEmailNotificationEmailAddress        DisplayString,
        rsEmailNotificationRowStatus           RowStatus
    }

rsEmailNotificationIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..10)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Index into the table"

    ::= { rsEmailNotificationConfigEntry 1 }


rsEmailNotificationAdminState OBJECT-TYPE
    SYNTAX      RsOAStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Administrative State of this row in the Email Notification table"

    ::= { rsEmailNotificationConfigEntry 2 }


rsEmailNotificationEmailAddress OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Email address to send the trap to."

    ::= { rsEmailNotificationConfigEntry 3 }
```

```
rsEmailNotificationRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Row status is used to create and delete rows in the table"

    ::= { rsEmailNotificationConfigEntry 4 }


-- End Table rsEmailNotificationConfigTable
=========================================================
rsUserInformation          OBJECT-IDENTITY
    STATUS    current
    DESCRIPTION
        "User information table"
    ::= { acttechnicoRAIDStor 7 }


rsAdminUserName OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Login name for the administrative user for RAIDStor"

    ::= { rsUserInformation 1 }

END
```

12 RAIDStor Software Revision History

The software revision on a RAIDStor can be determined by reading the /usr/share/snmp/raidstor.cfg file, or by examining the Blade Info page with a web browser. It consists of the date of the software release loaded on this blade and the "check-in" number from the revision control system used by ACT/Technico.

ACT uses "subversion" (SVN) for revision control for this project. On a software check-in the version number is automatically incremented. ACT tracks software versions by the date, followed by the SVN revision number. The RAIDStor software release was initially frozen on 03/03/06 at revision **02032006-202** for qualification testing. As a result of problems found during testing, a number of bug fixes and enhancements were made. The initial released software version was **09262006-237**.

The software repository originally contained software for several different RAIDStor projects using different hardware. As this particular project was fully qualified and released to production in 2007, an effort was made to remove all software not associated with this project to a new software repository to better isolate this project from other development efforts.

Below is a summary of the updates since the initial 237 release.

12.1 Software Revision Details

R238:

- Added customer specific job information to the database. It was later decided to store customer specific information outside the database so that the software distribution could be shared with customers.

R239:

- Added VME RAIDStor project files, later removed to new repository.

R240:

- Added more VME RAIDStor project files, again later removed.
- Updated version numbers in /etc/inittab and raidstor.cfg files.

R240:R241:R242:R243:R244:

- Removed files related to other projects from database.

R245:

- Deleted unused web pages.
- Updated the RAIDStor UserManual. The manual is still preliminary.

R246:

- Fixed status page problem on dual-blade RAIDStor.
- Modified shuttles script to remove wait_for_resync, this was causing boot to take an hour after a crash during local resync. This is definitly not needed on single blade.
- Deleted Lanpc3 software, this software is not supported.

R247:

- Removed files related to other projects from database.

R248:

- Update User Manual to V1.0. The RAIDStor User Manual shipped on every RAIDStor. It can be accessed from the Web Page.
- Update /etc/inittab, remove set of static date, enable gettys on RTM ports.
- Add dvdTest script for production testing.
- Add script to mount USB memory stick (mntcf) and cd (mntcd) for production testing.

R249:

- Update the RAIDStor User Manual to include a customer procedure to install software. The procedure requires a single file not under revision control, basically a tar-gzipped backup of the /raidstor directory from the RAIDStor production server.
- This release is missing the file get_hw_addr that is used by dual-blade RAIDStors. It was lost when heartbeat was last updated.
- The release is also missing the timezone patch for 2007 where the start/stop dates for daylight savings time changed.

R250:

- Deleted Timesys Linux files from this database. They are not used on this project.

R251:

- Deleted more files not used by this project.
- Updated the zoneinfo.tgz file used to install timezone information during an install.

R252:

- Updated RAIDStor User Manual with NFS performance numbers.
- Deleted additional TimeSys Linux files.

R253:

- Fixed the timezone patch installed with release 251. The wrong file was checked in.
- Linked /etc/localtime with EST5EDT. This file was linked with EASTERN previously. This is no longer valid.
- Added separate shuttles script for RAID 1+5 with wait_for_resync enabled for dual-blade only. This is required by DRBD.
- Added version 1.3 of the RAIDStor User Manual. Console cable pin-out changed from Rev – to Rev B. Manual is fully released and approved

R254:

- Deleted more files not used by this project.

R255:

- Fixed software installation bug. If a CF greater than 1GB in size is installed on the RAIDStor the /root partition will get installed on the CF instead of the first RAID partition /dev/md/0.

R256:

- 6211 “SATA” Shuttle updates to the RAIDStor User Manual.
- Revision history added to RAIDStor User Manual.
- Console cable wire colors changed due to different cable manufacturer.

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP