EtherDrive® LD/SR SATA+RAID Storage Appliance
Software User Manual
Update 6/22/09

1. Introduction

SR EtherDrive Storage is a block storage RAID appliance with front loading, hot-swappable SATA disk drives. Disks can be used individually or in RAID sets and are exported as logical storage devices accessible using the ATA-over-Ethernet (AoE) protocol. The appliance is built upon an embedded operating system similar to UNIX.

This manual explains the concepts behind and the commands used to manage the appliance. The output of commands listed in this document may differ from that output by your software. **Software functionality documented in this manual may not be available in firmware revisions dated older than this manual.** The command set for the LD appliance is identical to the SR appliance; the only difference between them is the form factor of the SATA disks. As a result, the LD appliance uses the same firmware as the SR appliance. This document does not cover the hardware used in constructing the appliance (please refer to the companion Installation Guide).

The terms client system and host system are used interchangeably to reference the computer accessing the SR storage over the Ethernet SAN. The term target is used to refer to SR storage LUNs accessed by the client system. The term initiator refers specifically to the host system AoE driver used to perform I/O to the SR target LUNs.

For quick start examples, please see **Appendix B**.

A low-traffic mailing list is available to notify customers of documentation and firmware updates. Please email support@coraid.com and ask to be placed on the SR announcement mailing list. SR firmware and documentation can be found at the SR support page: http://coraid.com/support/sr/.

2. Concepts

2.1. Network Connections

The Coraid SR appliance is equipped with two (or more) Ethernet ports and can be connected directly to a host system using Ethernet or to multiple host systems using standard Ethernet switching equipment. The SR presents all logical AoE devices to each segment.[†] For CEC and syslog configuration, ether0 and ether1 are the motherboard's onboard interfaces. Ether0 is the left interface and ether1 is the right interface, when viewed from the rear. For enumeration of additional interfaces, please see the SR Hardware Installation Guide.

2.2. Disks

Each SR appliance must be given a shelf address with the `shelf` command. Local disks are addressed as *<shelf>.<slot>*, where the slot address is the physical disk slot in the chassis. For example, if the shelf address were set to 0 the first ten disks would be addressed as *0.0, 0.1, ... 0.9.*

_____

[†] Administrators of SR420 and SR1520 models, please see the paper titled *SR Redundancy and Throughput in Linux* at the SR support page for performance considerations.

2.3. LUNs

The appliance exports LUNs to the network using the administrator assigned shelf address. LUNs are created containing one RAID and may be grown with additional RAIDs. Each RAID in a LUN may be of any size and type and is designated by a part number within the LUN. LUNs are grown by appending new component RAID parts. The creation of the LUN and its initial RAID is accomplished with the `make` command. Additional RAIDs may be added to a LUN with the `grow` command. From the host system on the network the SR looks like a shelf of LUNs. Addressing on the host system varies based on system dependent design rules. Please see documentation accompanying your selected AoE initiator driver for details.

One or more disks in the appliance can be exported individually as a typical JBOD configuration. To conform with the software architecture, a single disk element LUN is created by initialising a linear RAID over a single disk. The `jbod` command is provided to simplify this task. `Jbod` assumes the external LUN is the same as the internal disk slot being exported. If this is not desired, the administrator may create the single disk linear RAID LUN using the `make` command.

Specifying LUN RAID component devices is accomplished with a three tiered naming scheme: `lun.part.drive`. The `lun` is the Logical Unit Number exported, `part` is the RAID component in the LUN, and `drive` is the drive component in the RAID. As an example, `0.1.2` would be the third drive (`2`) in the second RAID part (`1`) of LUN `0`.

An administrator may choose to force LUNs to stop responding to requests. This is accomplished by changing the state of the LUN to offline using the `offline` command. To bring a LUN back online, an administrator can use the `online` command. All newly created LUNs are offline by default providing the administrator the opportunity to set the MAC mask list prior to bringing the LUN online.

2.4. Limiting Access

Administrators may limit client access to LUNs by giving each LUN a MAC mask list. By default, a LUN's mask list is empty permitting access to anyone on the network. MAC addresses can be added to a LUN's mask list using the `mask` command. If a LUN's mask list is not empty, only those MAC addresses in the mask list will be permitted access. The MAC mask list **must be set per LUN**.

2.5. RAID Spare Allocation

Spare allocation for redundant RAID levels is accomplished by two means. The administrator can choose to manage RAIDs manually by replacing failed drives using the `unfail` or `replace` commands. Alternately, the administrator can declare a pool of spares to be automatically allocated when a failure occurs using the `spare` command.

Spare allocation from the spare pool is accomplished with a best-fit algorithm. When a failure occurs, the spare pool is checked for the smallest drive that will satisfy the RAID's need. As an example, if the spare pool contained 500GB and 400GB drives and a failure occurred on a RAID needing a 300GB drive, a 400GB drive would be elected as the replacement.

When a RAID sustains a failure, a background process periodically checks for a spare to become available. If any redundant RAID is operating with a failure, adding a spare to an empty spare group will cause that spare to be allocated to the RAID with the failure. If the appliance sustains a disk failure to a RAID part that is currently having its redundancy reconstructed -- either raid5 parity rebuild or mirror rebuild -- a double failure will occur causing the appliance to mark the RAID and its associated LUN as unusable. The LUN must be removed and remade to be reinstated. Please contact Coraid support for assistance with properly reinstating failed arrays.

There is a single spare pool for all RAIDs managed within the appliance.

2.6. Coraid Ethernet Console (CEC)

CEC is a method for obtaining a console connection to the SR appliance using standard Ethernet frames. CEC is very lightweight and does not include any security or encryption mechanisms. The commands `cecon` and `cecoff` are used to manage which interfaces accept CEC connections.

For ease of first configuration, CEC is enabled on the onboard motherboard interfaces on new shipments. Shelves without an assigned shelf address will show up as shelf address -1 to a CEC client.

CEC first appeared in SR release 20060717; users updating to obtain CEC functionality must manually enable CEC. **Be aware that enabling CEC and failing to logout may leave your system insecure depending on your network configuration.** As with AoE, the appliance is only as secure as your network.

The CEC configuration will persist across reboot. To obtain a CEC client, please visit the SR support page at coraid.com.

2.7. Shutting Down the System

Administrators should use the `halt` or `reboot` command to cleanly shut down the appliance. Both commands flush out dirty buffers, mark all RAIDs as cleanly shut down, and bring the system to a halt. Reboot will then reboot the system, while halt will await user input to reboot. As a matter of policy, when the appliance starts it will validate all raid5 and raid6rs RAIDs that are not marked as clean, ensuring the parity is correct.

2.8. RAID Configuration

Each RAID's configuration is saved on the component drives used in the RAID. At shutdown, the system must be able to communicate with all of a RAID's components in order to cleanly release the RAID.

2.9. Specifying Devices

Many commands require the specification of one or more LUNs or disk drives. To ease the specification of these devices a special range character (`-`) may be used. As an example, `0.2-5` is a valid substitution for `0.2 0.3 0.4 0.5`, indicating disks 2 through 5 on shelf 0. Analagously for LUN, `0-3` is a valid substitution for `0 1 2 3`, indicating LUNs 0 through 3.

2.10. Console Port Settings

In addition to using CEC and keyboard and video display, the Coraid SR appliance can be configured using the serial port. Before connecting a terminal to the serial port, configure the terminal to match the serial port: **9600 8-N-1** (9600 baud, 8 data bits, no parity, 1 stop bit).

2.11. Flash Signature

As of release 20071031, all SR flash boot modules must be signed in order for disk backed LUNs to be brought online. SR appliances shipped after this date are presigned, but users running older firmware will need to update their firmware and contact Coraid support with the output of `sigcheck -l` to obtain a valid signature.

2.12. SMART Support

SMART (Self Monitoring, Analysis, and Reporting Technology) support is limited to reporting the drive's internal SMART status. A SATA drive continually monitors various internal counters and determines when a threshhold has been exceeded for which drive replacement is considered necessary. The SMART status can be obtained with the `show -s` command. Additionally, the appliance will itself retrieve this information once an hour and upon receipt of a threshhold exceeded condition, send a message via syslog. The SR will continue to generate this syslog message every 6 hours for the first 24 hours, then every 24 hours thereafter, until the user disables SMART on the drive with the `smartdisable` command. It should be noted that some hard drives are shipped with SMART disabled by default and may need to have SMART enabled with the `smartenable` command.

2.13. Keyboard Shortcuts

In order to make the system more friendly to users accustomed to the GNU readline interface, the following control sequences are provided. A history buffer of 32 lines is maintained.

backspace	erase the character before the cursor
ˆd	delete character under cursor
ˆw	delete previous word
ˆu	delete entire line
ˆb	cursor back one char
ˆf	cursor forward one char
ˆa	cursor beginning of line
ˆe	cursor end of line
ˆk	kill text from cursor to end of line
ˆy	yank killed text into line after cursor
ˆp	go back one line in the history, cursor at end of line
ˆn	go forward one line in the history, cursor at end of line

In addition, the control sequences emitted by most Linux systems for the left, right, up, and down arrows are recognized. These are equivalent to ˆb, ˆf, ˆp and ˆn, respectively.

2.14. RAIDShield™ Protection

Each redundant RAID is protected by a mechanism called the RAIDShield. When a disk in a well-functioning -- not degraded or failed -- redundant RAID exhibits a block read error, the RAIDShield is enacted to calculate the block contents from the remaining disks in the array. RAIDShield then writes the block to the "failed" disk and then rereads it to permit the disk to internally remap the bad sector(s). Whether or not this is successful, the user will be notified by a syslog diagnostic of the attempt. As disks become ever larger the likelihood of sector read errors increases. RAIDShield corrections do not necessarily indicate a drive needs to be replaced.

A special case occurs if the RAIDShield is enacted to recover from a failure on a disk in a raid5 or raid6rs when the parity is not yet initialized. This can occur when the RAID is first created, but is more common when the appliance goes down without cleanly releasing the RAID. In the latter case the parity for the row containing the failure is likely to be correct and can be relied upon to reconstruct the block. Since this cannot be guaranteed, however, RAIDShield will take the LUN offline to protect the user from potential data corruption. The user must acknowledge the potential error by running the online command twice to bring the LUN back online. As expected, syslog messages are generated at every step.

Also covered under the RAIDShield is a background scrubber process that continually reads redundant RAID arrays to detect and correct read failures using the same mechanism described above. The RAIDShield scrubber operates in a loop reading a predetermined amount of data and sleeping for one second. The amount of data read is determined based on the access of the RAID array and is calculated to avoid conflicting with user access I/O. The scrubber's access pattern will cause the disk access LEDs on scrubbed disks to flash every second.

The scrubber will not run if the array is initing, degraded, or failed. As a result of the scrubber process, disks in redundant arrays are rarely idle. As disk access does not contribute to shortened disk life this should not be a concern.

RAIDShield protection extends to user notification of latent failures of spare drives. Once a minute RAIDShield reads from the beginning of all spare drives to verify they are accessible. If this check fails, a syslog message is sent to notify the user of the potentially dead spare drive. Spares are not scrubbed like other RAID elements; when they are allocated every block is written permitting the drive to remap any bad spots.

2.15. SES-2 Status and Control

SES-2 is a protocol used for control and status of elements in a storage enclosure. Coraid's SR16xx, SR24xx, and LD24xx[1] appliances come equipped with controllers on the SATA backplane that provide this capability. Currently SES-2 communication is limited to these models.

SES-2 can be used to query the SATA backplane for the status of an audible alarm, a temperature sensor, and red indicator LEDs at each slot. Additionally this communication path can be used to set each slot's LED state and manipulate the audible alarm. For more information, please see the commands `alarm`, `slotled`, `temp`, and `sesstat`. The SR uses the `slotled` command to automatically set the slot LED state in accordance with the RAID configuration.

_____

[1] Fan status is not available on the LD Appliance due to chassis design.

3. Commands

Commands are presented in the following sections in the order that most facilitates learning to use the appliance. The following table lists the commands in alphabetical order for quick lookup once the command set has been reviewed.

alarm(3.4.2)	jbod(3.3.16)	replace(3.3.12)	smartenable(3.2.2)
cecoff(3.1.9)	list(3.3.1)	resetdisk(3.2.4)	sos(3.1.2)
cecon(3.1.8)	make(3.3.2)	restore(3.3.14)	spare(3.3.9)
eject(3.3.13)	mask(3.3.8)	rmspare(3.3.10)	syslog(3.1.6)
exit(3.1.10)	offline(3.3.7)	sesstat(3.4.4)	temp(3.4.3)
fail(3.3.11)	online(3.3.6)	shelf(3.1.5)	update(3.3.15)
grow(3.3.3)	passwd(3.1.3)	show(3.2.1)	when(3.3.4)
halt(3.1.12)	reboot(3.1.11)	sigcheck(3.1.13)	
help(3.1.1)	release(3.1.4)	slotled(3.4.1)	
ifstat(3.1.7)	remove(3.3.5)	smartdisable(3.2.3)	

3.1. Base System Configuration and Use

3.1.1. help

```
usage: help [ cmd ]
```

The `help` command prints the list of user commands to the console. Specifying a `cmd` argument displays the usage for the named command.

3.1.2. sos

```
usage: sos
```

The `sos` command prints diagnostic information to send to Coraid support. The output contains the SR release, model, sata card rev, disk information (model/serial/firmware), and LUN/raid configuration.

Providing the output of this command when first contacting support will often result in the fastest solution to a problem.

3.1.3. passwd

```
usage: passwd
```

The `passwd` command sets the console login password. If the administrator should lose the password given to the appliance, entering the reserved password `ivelostit` will begin a challenge/response dialog which the administrator must contact Coraid support to complete. After successfully passing the challenge/response dialog, the password will be cleared.

3.1.4. release

```
usage: release
```

The `release` command prints the release date of the currently running firmware.

Example:

```
SR shelf 1> release
20080304 - Tue Mar 4 15:46:52 EST 2008
SR shelf 1>
```

3.1.5. shelf

```
usage: shelf [ shelfno ]
```

The `shelf` command sets the shelf address of the appliance. The argument to shelf should be a value between 0 and 65534, inclusive. It is recommended that the administrator ensure each shelf address is unique among all AoE storage devices attached to the network. Without an argument, the shelf address is printed.

Example:

```
SR shelf 1> shelf 2
SR shelf 2> shelf
2
SR shelf 2>
```

3.1.6. syslog

```
usage: syslog [ -cp ] [ -s severity ] message
```

The `syslog` command sends syslog messages to a remote syslog host at UDP port 514. Given the `-c` flag, `syslog` enters a dialogue for the administrator to specify the source and destination IP used in the syslog datagram, as well as the SR local interface to use when sending syslog messages. The destination IP should be set to the desired syslog host. The administrator need only set this configuration once; future calls to syslog will use the stored information. Given the `-p` flag, `syslog` prints the syslog configuration set with `-c`. The `-s` flag requires an integer argument and sets the syslog severity for this message only.

If unset, the source IP address used in the syslog UDP datagram is 205.185.197.30. The syslog facility used is *local0 (16)* and if unspecified, the severity is *informational (6)*. The SR generates messages using the default severity.

Syslog messages are only sent out one interface to avoid duplication in certain network configurations. By default, the first interface (ether0) is used.

Configuration of the syslog host is beyond the scope of this document.

`Syslog` always prints its message to the console.

Example:

```
SR shelf 1> syslog -c
Configuring syslog.  Enter IP addresses in dotted notation.
Local interface is in the format ether[0-9].
IPv4 destination address []: 192.168.0.1
IPv4 source address [205.185.197.30]: 192.168.0.30
Local syslog interface [ether0]: ether1
Configuration successful.
SR shelf 1> syslog -p
 destination IP: 192.168.0.1
      source IP: 192.168.0.30
local interface: ether1
SR shelf 1>
```

3.1.7. ifstat

```
usage: ifstat [ -a ] [ interface ... ]
```

The ifstat command displays the status of the Ethernet interfaces. The Ethernet address, link speed, and capable MTU are reported. The Ethernet MTU is not configurable. Without an argument, ifstat lists all system interfaces. The interface specification is of the form ether[0-9]. The -a flag displays verbose statistics about the interface(s).

Example:

```
SR shelf 1> ifstat
  name            addr    link (Mbps)     mtu
ether0   00304833f674    1000/ 1000     9014
ether1   00304833f675    1000/ 1000     9014
ether2   0060dd4752e6       0/10000     9000
SR shelf 1>
```

3.1.8. cecon

```
usage: cecon [ interface ]
```

The cecon command enables CEC for a specified interface. Without an argument, cecon lists all interfaces for which CEC is served. The interface specification is of the form ether[0-9].

Example:

```
SR shelf 1> cecon ether1
SR shelf 1> cecon
/net/ether1
SR shelf 1>
```

3.1.9. cecoff

```
usage: cecoff interface
```

The cecoff command disables CEC on a specified interface. The interface specification is of the form ether[0-9].

Example:

```
SR shelf 1> cecon
/net/ether1
SR shelf 1> cecoff ether1
SR shelf 1> cecon
SR shelf 1>
```

3.1.10. exit

```
usage: exit
```

The exit command exits the command line interface. If a password has been set with the passwd command, the login prompt is issued.

3.1.11. reboot

```
usage: reboot
```

The reboot command cleanly shuts down all LUNs and their component RAIDs and reboots the system.

3.1.12. halt

usage: `halt`

The `halt` command cleanly shuts down all LUNs and their component RAIDs and halts the system. After the system is halted pressing enter reboots the unit.

3.1.13. sigcheck

usage: `sigcheck [ -ls signature ]`

The `sigcheck` command lists and sets the signature on the SR boot flash. A signature is unique to the SR boot flash for which it is generated. The `-l` flag displays the signature details. The output of `sigcheck` `-l` can be provided to Coraid support in order to obtain a flash signature. The `-s` flag accepts a signature parameter and directs sigcheck to validate the signature. If the signature is valid, sigcheck stores the signature on the SR boot flash for future use.

3.2. Disk Status and Use

3.2.1. show

```
usage: show [ -lcms ]
```

The show command displays information about the disks in the appliance.

The -l (the letter "l") flag shows extended information about each disk, including its size and state. Valid disk states are **missing**, **initing**, **up**, and **connectfail**. If the **connectfail** disk state is listed, the SR has given up trying to initialize the disk. This state is typically only entered when a disk is misbehaving; the SR stops communicating with the disk to conserve system resources. The user must use the resetdisk command to manually reset the disk and retry initialization.

The -m flag reports the SATA connection mode; valid states are **sata1**, **sata2**, or **missing**. SATA connection modes are determined by autonegotiation and are not controllable. It is common for disks in an SR to be in a mixed sata1/sata2 state, even for identical disk models.

The -s flag reports the SMART status; valid states are **normal**, **threshhold exceeded**, **disabled**, and **unknown**. The unknown condition is provided when a disk is missing or SMART communication is unsuccessful.

The -c flag shows raid configuration information sometimes needed by coraid support.

Example:

```
SR shelf 1> show -l
1.0      82.35GB up
1.1      82.35GB up
1.2      82.35GB up
1.3      82.35GB up
SR shelf 1> show -s
1.0    normal
1.1    normal
1.2    normal
1.3    normal
SR shelf 1>
```

3.2.2. smartenable

```
usage: smartenable shelf.slot ...
```

The smartenable command enables the SMART command feature set on a drive. SMART must be enabled in order for the SMART status to be obtainable with show -s. This state is maintained by the disk and is persistent across power cycle.

3.2.3. smartdisable

```
usage: smartdisable shelf.slot ...
```

The smartdisable command disables the SMART command feature set on a drive. This is primarily useful when the administrator no longer wants to be notified about a disk that has exceeded an error threshhold.

3.2.4. resetdisk

```
usage: resetdisk shelf.slot ...
```

The resetdisk command manually resets the SATA connection to one or more disk slots. It is only useful when a disk has entered into the **connectfail** state.

3.3. LUN/RAID Configuration and Use

3.3.1. list

```
usage: list [ -l ] [ lun ... ]
```

The list command shows the LUNs currently exported by the appliance. If a lun is not specified, all current LUNs are listed. Each line output lists the LUN, size, and state. A LUNs's state is either online or offline, indicating whether or not it is accepting requests.

The -l (the letter "l") flag gives detailed information about the LUN's components. The first line in each record displays the LUN, size, and state of the LUN. The next indention level displays component RAID parts and their size, type, state, and during parity initialization/reconstruction, percentage completion. The state of the component RAID is one or more of:

initing - RAID is initializing parity.
recovering - RAID is rebuilding replaced component.
degraded - RAID is operating with failed or missing components.
failed - RAID has sustained too many component failures and is unusable.
normal - RAID is operating normally.

Following the RAID component at the next indention level are the drive components in the RAID, one per line. Each line displays the drive's RAID component address, state, size, and physical shelf.slot location. The state of the drive component is one of:

failed - drive has failed.
replaced - drive is being used as a replacement for a failed drive.
missing - a placeholder for a missing drive; possible if all components are not available at startup time.
normal - drive is operating normally.

Example:

```
SR shelf 1> list
 0   82.348GB online
 1   82.348GB online
 2   82.348GB online
 3   82.348GB online
SR shelf 1> list -l
 0   82.348GB online
  0.0    82.348GB raidL
    0.0.0  normal      82.348GB 1.0
 1   82.348GB online
  1.0    82.348GB raidL
    1.0.0  normal      82.348GB 1.1
 2   82.348GB online
  2.0    82.348GB raidL
    2.0.0  normal      82.348GB 1.2
 3   82.348GB online
  3.0    82.348GB raidL
    3.0.0  normal      82.348GB 1.3
SR shelf 1>
```

3.3.2. make

```
usage: make lun raidtype [ shelf.slot ... ]
```

The make command creates RAIDs and initializes them to a LUN.

The LUN field must be a value between 0 and 63, inclusive. If set sufficiently high the LUN may not be addressible from the client system. Coraid recommends administrators verify LUN addressibility with host system drivers prior to allocation.

The raidtype field may be one of:

raidL	A linear raid device
raid0	A striped raid device
raid1	A mirrored raid device
raid5	A round-robin parity raid device
raid6rs	A double fault tolerant round-robin parity raid device using Reed-Solomon syndromes
raid10	A stripe of mirrors raid device
raw	A raw export of an underlying device
update	A ram based device for appliance update

A full description of the various RAID levels is beyond the scope of this document. The raw raidtype is a simple block for block export of a single disk. Since no portion of the disk is used for raid configuration storage, the LUN will not persist across reboot. The update raidtype is a pseudo type for a raw device over a RAM based disk. No component drives should be specified when declaring the update LUN. Please see **Appendix C** for an explanation of the software update procedure.

The shelf.slot fields following raidtype specify the drives to be used as components of the RAID.

The make command initializes a LUN in the offline state, permitting administrators to set a mac mask list prior to making the LUN available for access. Generally, newly created LUNs must be brought online with the online command before they will be visible to host systems. There are two exceptions to this rule; the update LUN and LUNs created with the jbod command are automatically brought online as a convenience.

For raid10, the drive list must contain an even number of elements. The list is split into two equal ordered sets at the center of the list. Mirrors are constructed across the pairs of drives in the same position in each set. Once the mirrors are chosen, a stripe is placed across all mirrored elements.

In the example below, the mirrored elements of LUN 1 are {1.4, 1.7}, {1.5, 1.8}, and {1.6, 1.9}.

Example:

```
SR shelf 1> make 1 raid10 1.4-9
SR shelf 1> list
 1  247.045GB offline
SR shelf 1> online 1
SR shelf 1> list -l
 1  247.045GB online
  1.0    247.045GB raid10
    1.0.0  normal       82.348GB 1.4
    1.0.1  normal       82.348GB 1.5
    1.0.2  normal       82.348GB 1.6
    1.0.3  normal       82.348GB 1.7
    1.0.4  normal       82.348GB 1.8
    1.0.5  normal       82.348GB 1.9
SR shelf 1>
```

3.3.3. grow

```
usage: grow lun raidtype shelf.slot ...
```

The grow command adds a RAID to an existing LUN. The command syntax is identical to that of the make command. Once added to LUN, a RAID may not be removed. Storage associated with LUN can only be reclaimed by removing the entire LUN with the remove command.

Example:

```
SR shelf 1> list -l
 0 1500.324GB online
  0.0   1500.324GB raid5 initing  0.06%
    0.0.0  normal    500.108GB 1.0
    0.0.1  normal    500.108GB 1.1
    0.0.2  normal    500.108GB 1.2
    0.0.3  normal    500.108GB 1.3
SR shelf 1> grow 0 raid5 1.4-7
beginning building parity: 0.1
SR shelf 1> list -l
 0 3000.647GB online
  0.0   1500.324GB raid5 initing  0.14%
    0.0.0  normal    500.108GB 1.0
    0.0.1  normal    500.108GB 1.1
    0.0.2  normal    500.108GB 1.2
    0.0.3  normal    500.108GB 1.3
  0.1   1500.324GB raid5 initing  0.02%
    0.1.0  normal    500.108GB 1.4
    0.1.1  normal    500.108GB 1.5
    0.1.2  normal    500.108GB 1.6
    0.1.3  normal    500.108GB 1.7
SR shelf 1>
```

3.3.4. when

```
usage: when
```

The when command lists RAID devices in the initing or recovering state and shows their I/O rate and estimeted time to completion. Time is formatted as hours:minutes:seconds.

Example:

```
SR shelf 1> when
0.0 90701 KBps 0:55:18 left
SR shelf 1>
```

3.3.5. remove

```
usage: remove lun ...
```

The remove command removes one or more LUNs. All drives used in component RAIDs are released for reuse.

Example:

```
SR shelf 1> list -l
 0 1500.324GB online
  0.0   1500.324GB raid5 initing  0.37%
    0.0.0  normal    500.108GB 1.0
    0.0.1  normal    500.108GB 1.1
    0.0.2  normal    500.108GB 1.2
    0.0.3  normal    500.108GB 1.3
 1 1000.216GB online
  1.0   1000.216GB raid10 normal
    1.0.0  normal    500.108GB 1.4
    1.0.1  normal    500.108GB 1.5
    1.0.2  normal    500.108GB 1.6
    1.0.3  normal    500.108GB 1.7
SR shelf 1> remove 0 1
Removing lun(s): 0 1
building parity aborted: 0.0
SR shelf 1>
```

3.3.6. online

```
usage: online [ lun ... ]
```

The online command moves one or more LUNs into the online state, enabling them for network access. Without an argument, online lists all LUNs currently online.

Example:

```
SR shelf 1> online 0
SR shelf 1> online
 0    82.35GB online
SR shelf 1>
```

3.3.7. offline

```
usage: offline [ lun ... ]
```

The offline command moves one or more LUNs into the offline state. While offline, LUNs are not accessible from the network. Without an argument, offline lists all LUNs currently offline.

Example:

```
SR shelf 1> offline 0
SR shelf 1> offline
 0    82.35GB offline
SR shelf 1>
```

3.3.8. mask

```
usage: mask [ lun ... [ +mac ... ] [ -mac ... ] ]
```

The mask command manages client access to exported LUNs. Without any arguments, mask lists all LUNs and their mask lists. Given only LUN arguments, mask lists the mac mask list for all specified LUNs. Further arguments are directives to add (+) or remove (−) mac addresses from the mask list.

Example:

```
SR shelf 1> mask
0 000E0C65BAA3
SR shelf 1> mask 0 +000E0C65BAA2 +000E0C65BAA4 +000E0C65BAA5
SR shelf 1> mask
0  000E0C65BAA5 000E0C65BAA4 000E0C65BAA2 000E0C65BAA3
SR shelf 1> mask 0 -000E0C65BAA4
SR shelf 1> mask
0  000E0C65BAA5 000E0C65BAA2 000E0C65BAA3
SR shelf 1> mask 0 -000E0C65BAA5 -000E0C65BAA2 -000E0C65BAA3
SR shelf 1> mask
0
SR shelf 1>
```

3.3.9. spare

```
usage: spare [ shelf.slot ... ]
```

The spare command lists and manages the spare device pool. For each device listed, spare attempts to recruit the device as a spare. If the device is already in use, an error will occur. Without arguments, spare lists all devices in the spare pool.

Example:

```
SR shelf 1> spare 1.0-1
SR shelf 1> spare
1.0  500.108GB
1.1  500.108GB
SR shelf 1>
```

3.3.10. rmspare

```
usage: rmspare shelf.slot ...
```

The rmspare command removes devices from the spare pool. One or more devices may be specified as arguments.

Example:

```
SR shelf 1> spare
1.0  500.108GB
1.1  500.108GB
SR shelf 1> rmspare 1.0
SR shelf 1> spare
1.1  500.108GB
SR shelf 1>
```

3.3.11. fail

`usage: fail lun.part.drive`

The `fail` command changes the state of a drive to failed. It is a convenient method for testing appliance behaviour in failure conditions.

Example:

```
SR shelf 1> list -l
 0  500.108GB online
  0.0    500.108GB raid1 normal
    0.0.0  normal    500.108GB 1.0
    0.0.1  normal    500.108GB 1.1
SR shelf 1> fail 0.0.1
SR shelf 1> list -l
 0  500.108GB online
  0.0    500.108GB raid1 degraded
    0.0.0  normal    500.108GB 1.0
    0.0.1  failed    500.108GB 1.1
SR shelf 1>
```

3.3.12. replace

`usage: replace lun.part.drive shelf.slot`

The `replace` command replaces a failed component in a redundant RAID (raid5, raid1, raid10) with a new drive. After replacement, the drive will be reconstructed. The specified drive must be listed as up in show -l output and cannot be in use as a spare or RAID component, with one exception. It is legal to replace a RAID component with itself to force recovery of the existing component drive. The following example shows this usage.

Example:

```
SR shelf 1> list -l
 0  500.108GB offline
  0.0    500.108GB raid1 degraded
    0.0.0  normal    500.108GB 1.0
    0.0.1  failed    500.108GB 1.2
SR shelf 1> replace 0.0.1 1.1
SR shelf 1> list -l
 0  500.108GB offline
  0.0    500.108GB raid1 recovering,degraded  0.05%
    0.0.0  normal    500.108GB 1.0
    0.0.1  replaced  500.108GB 1.1
SR shelf 1> fail 0.0.1
SR shelf 1> list -l
 0  500.108GB offline
  0.0    500.108GB raid1 degraded
    0.0.0  normal    500.108GB 1.0
    0.0.1  failed    500.108GB 1.1
SR shelf 1> replace 0.0.1 1.1
SR shelf 1> list -l
 0  500.108GB offline
  0.0    500.108GB raid1 recovering,degraded  0.14%
    0.0.0  normal    500.108GB 1.0
    0.0.1  replaced  500.108GB 1.1
SR shelf 1>
```

3.3.13. eject

```
usage: eject lun ...
```

The eject command is very similar to the remove command, except eject will not clear the raid configuration on the LUN's component drives. This is primarily useful when the administrator wants to move a LUN from one shelf to another without shutting down.

```
SR shelf 1> list -l
 0   500.108GB offline
  0.0     500.108GB raid1 normal
    0.0.0  normal     500.108GB 1.4
    0.0.1  normal     500.108GB 1.5
SR shelf 1> eject 0
Ejecting lun(s): 0
SR shelf 1>
```

3.3.14. restore

```
usage: restore [ -l ] [ oldshelf [ oldlun [ newlun ] ] ]
```

The restore command reads the SR disk configuration from all drives in the system and builds LUNs and spares. The restore command ignores any disks that do not belong to the configured shelf address of the SR. This command is run on boot by initialization software without any arguments to start the system. Given the -l flag, restore displays the commands to be run and exits without performing any action.

Restore can be used to import LUNs and spares that once resided on another shelf by providing the old-shelf argument. To pull in only a specific LUN, the oldlun argument can be used; the resulting LUN will be identical to oldlun. If this is undesirable, a further newlun argument can be provided to specify a new LUN.

If run without arguments, restore may attempt to create LUNs that already exist. Redeclaration of a LUN will be met with an error message.

Please note that restore -l may display undocumented formats and flags that the user is insulated from needing to know.

```
SR shelf 1> list -l
SR shelf 1> restore -l
Reading config information from drives ... done.
make -r 0 raid1 1.4:c 1.5:c
online 0
SR shelf 1> restore
Reading config information from drives ... done.
SR shelf 1> list -l
 0  500.108GB online
  0.0     500.108GB raid1 normal
    0.0.0  normal     500.108GB 1.4
    0.0.1  normal     500.108GB 1.5
SR shelf 1>
```

3.3.15. update

```
usage: update [ -l ]
```

The `update` command processes the **tarc** file on the update LUN. With the `-l` (the letter "l") flag `update` displays the LUN being used as the update target without performing any action. For an explanation of the software update procedure, please see **Appendix C**.

Example:

```
SR shelf 1> update -l
0
SR shelf 1> update
Updating kernel ... done.
Updating root fs ... done.
Update successful. Please reboot to use the new software.
SR shelf 1>
```

3.3.16. jbod

```
usage: jbod shelf.slot ...
```

The `jbod` command exports one or more drive slots as LUNs. It is provided as a convenience for administrators wishing to easily obtain a JBOD configuration. All LUNs created by `jbod` are automatically brought online.

`Jbod` simply makes a linear RAID LUN over a single shelf.slot, setting the LUN to the drive slot. Administrators may choose to export the drives as other LUNs by creating the raidL LUN themselves. As an example, the following two sets of commands are identical in their effect:

```
SR shelf 1> jbod 1.0
```

```
SR shelf 1> make 0 raidL 1.0
SR shelf 1> online 0
```

For an example of configuring the appliance as a JBOD, please see **Appendix B**.

3.4. SES-2 Status and Control

3.4.1. slotled

`usage: slotled slot [ ident | fault | rebuild | spare | reset ]`

The `slotled` command interacts with the red indicator LEDs at each drive slot. Each indicator LED has an independent state that controls the LED behaviour. The LED states are as follows:

State	Description	LED behaviour
ident	slot identify/locate	FAST BLINK (4Hz)
fault	slot is failed	SOLID ON, audible alarm triggered
rebuild	slot is rebuilding	SLOW BLINK (1Hz)
spare	slot is a hot spare	2 FAST BLINK, PAUSE [repeats]
reset	slot has no state	OFF

Without an LED directive, `slotled` reports the LED state for the specified slot. Physical ejection/insertion of the disk resets the slot LED state. The audible alarm is active if any LED is in the `fault` state. The SR uses the `slotled` command to automatically set the slot LED state in accordance with the RAID configuration.

Example:

```
SR shelf 1> slotled 0
slot0: reset
SR shelf 1> slotled 0 ident
SR shelf 1> slotled 0
slot0: ident
SR shelf 1> slotled 0 reset
SR shelf 1> slotled 0
slot0: reset
SR shelf 1>
```

3.4.2. alarm

`usage: alarm [ on | mute | reset ]`

The `alarm` command interacts with the audible alarm on the SATA backplane. Without an argument, the alarm state is displayed. The audible alarm is enabled by a fan failure/ejection, or by the LED state of any slot being in the fault state. The alarm state is either on (audible) or reset (off, ready to be tripped by failure), or muted.

Example:

```
SR shelf 1> alarm
alarm: reset
SR shelf 1> alarm
alarm: on
SR shelf 1> alarm mute
SR shelf 1> alarm
alarm: muted
SR shelf 1> alarm reset
SR shelf 1> alarm
alarm: reset
SR shelf 1>
```

3.4.3. temp

```
usage: temp
```

The temp command displays the temperature reported by the sensor on the SATA backplane.

Example:

```
SR shelf 1> temp
temp: 30C
SR shelf 1>
```

3.4.4. sesstat

```
usage: sesstat
```

The sesstat command simply reports the SES-2 status of all the components at the SATA backplane by running slotled for each disk slot, and the alarm, and temp commands.

Appendix A -- Support

Coraid support for the SR appliance is accomplished via email through `support@coraid.com`. Since our line of appliances is primarily command line based, text-based support enables us to isolate problems and clearly communicate solutions. This is uncomfortable for users at first as we've all been subject to poor email support and autogenerated responses. Coraid takes great pride in our level of customer support, providing top level engineers as first responders. In order to make this interaction most effective it is important to provide detailed information about the problem at hand. In many cases, our engineers can provide a solution in the first response.

A good first step is to check the SR FAQ at the SR support page before contacting Coraid support to see if the question has already been answered.

For SR problems, include the output of the `sos` command in the first email to support. This information gives us a snapshot of your SR and in many cases will dictate the problem at hand.

Use the serial console or CEC (the Coraid Ethernet Console) to get real text to send us instead of summarizing what's going on. For example, the output of "list -l" is much more helpful than saying something like, "list -l shows the same thing."

Include a description of how the networking is configured. For simplicity in eliminating potential problems, it's best to start with a single cable between the SR and the host system. The SR has automatic MDI/MDI-X capability and will auto-negotiate crossover.

Our high quality email support routinely converts customers who are accustomed to tiered phone support. Email support doesn't require a complete halt to all other activities -- a must in today's multitasking work environment. Customers also enjoy being able to have past solutions easily referenced from their inbox. It can take a little getting used to, but in our experience it's a big win for us and for our customers. We sincerely hope you'll agree.

Appendix B -- Quick Start Examples

The following example initializes a raid5 over 14 disks in the appliance. The remaining disk is allocated as a spare.

```
SR shelf 1> show -l
   1.0      82.35GB up
   1.1      82.35GB up
   1.2      82.35GB up
   1.3      82.35GB up
   1.4      82.35GB up
   1.5      82.35GB up
   1.6      82.35GB up
   1.7      82.35GB up
   1.8      82.35GB up
   1.9      82.35GB up
   1.10     82.35GB up
   1.11     82.35GB up
   1.12     82.35GB up
   1.13     82.35GB up
   1.14     82.35GB up
SR shelf 1> list
SR shelf 1> make 0 raid5 1.0-13
SR shelf 1> spare 1.14
SR shelf 1> online 0
SR shelf 1> list -l
 0 1070.527GB online
  0.0   1070.527GB raid5 initing 0.02%
    0.0.0  normal       82.348GB 1.0
    0.0.1  normal       82.348GB 1.1
    0.0.2  normal       82.348GB 1.2
    0.0.3  normal       82.348GB 1.3
    0.0.4  normal       82.348GB 1.4
    0.0.5  normal       82.348GB 1.5
    0.0.6  normal       82.348GB 1.6
    0.0.7  normal       82.348GB 1.7
    0.0.8  normal       82.348GB 1.8
    0.0.9  normal       82.348GB 1.9
    0.0.10 normal       82.348GB 1.10
    0.0.11 normal       82.348GB 1.11
    0.0.12 normal       82.348GB 1.12
    0.0.13 normal       82.348GB 1.13
SR shelf 1> spare
1.14   82.348GB
SR shelf 1>
```

The following example configures the appliance as a JBOD.

```
SR shelf 1> jbod 1.0-14
SR shelf 1> list -l
 0   82.348GB online
  0.0      82.348GB raidL
    0.0.0  normal      82.348GB 1.0
 1   82.348GB online
  1.0      82.348GB raidL
    1.0.0  normal      82.348GB 1.1
 2   82.348GB online
  2.0      82.348GB raidL
    2.0.0  normal      82.348GB 1.2
 3   82.348GB online
  3.0      82.348GB raidL
    3.0.0  normal      82.348GB 1.3
 4   82.348GB online
  4.0      82.348GB raidL
    4.0.0  normal      82.348GB 1.4
 5   82.348GB online
  5.0      82.348GB raidL
    5.0.0  normal      82.348GB 1.5
 6   82.348GB online
  6.0      82.348GB raidL
    6.0.0  normal      82.348GB 1.6
 7   82.348GB online
  7.0      82.348GB raidL
    7.0.0  normal      82.348GB 1.7
 8   82.348GB online
  8.0      82.348GB raidL
    8.0.0  normal      82.348GB 1.8
 9   82.348GB online
  9.0      82.348GB raidL
    9.0.0  normal      82.348GB 1.9
10   82.348GB online
  10.0    82.348GB raidL
    10.0.0 normal      82.348GB 1.10
11   82.348GB online
  11.0    82.348GB raidL
    11.0.0 normal      82.348GB 1.11
12   82.348GB online
  12.0    82.348GB raidL
    12.0.0 normal      82.348GB 1.12
13   82.348GB online
  13.0    82.348GB raidL
    13.0.0 normal      82.348GB 1.13
14   82.348GB online
  14.0    82.348GB raidL
    14.0.0 normal      82.348GB 1.14
SR shelf 1>
```

Appendix C -- Updating

Updating the SR is accomplished by a three step process:

- Make a RAM based update LUN using `make`.
- Copy the appropriate **tarc** update file to the update LUN from a host system.
- Run the update command on the appliance to process the **tarc** update.

The **tarc** update file contains a kernel update and a root filesystem update. Command updates are performed as a process of updating the flash root filesystem.

The following example performs the above three steps to update an SR. The host system used is Linux with a 2.6 kernel. Recent versions of aoe-tools on Linux (bundled with the standalone AoE driver) provide the `coraid-update` utility that performs simple checks to validate the destination target and the tarc update file. Users are recommended to use this utility to avoid simple mistakes in using dd to copy the file to the update LUN. Updating from a client system (solaris, freebsd, etc) lacking `coraid-update` is accomplished with `dd`; simply write the update tarc file to the raw block device represented by the update LUN.

Please use care to write the tarc update file to the update LUN only. Writing the tarc update file to any LUNs backed by disk storage will cause data loss.

Example:

```
SR shelf 1> make 15 update
SR shelf 1> update -l
15
SR shelf 1> list -l
 15    0.041GB online
  15.0      0.041GB raw normal
    15.0.0  normal        0.041GB update
SR shelf 1>
```

On the client system:

```
% modprobe aoe
% aoe-stat
    e1.15         0.040GB  eth4,eth5 1024  up
% # Verify the md5sum with that reported by the SR support page
% md5sum SR20080407.tarc
b1936c99efe79a41204eb0f4a58192db
% coraid-update SR20080407.tarc /dev/etherd/e1.15
% # Or alternately, if your system lacks coraid-update
% dd if=SR20080407.tarc of=/dev/etherd/e1.15
5260+0 records in
5260+0 records out
%
```

Back on the SR appliance:

```
SR shelf 1> update
Updating kernel ... done.
Updating root fs ... done.
Update successful. Please reboot to use the new software.
SR shelf 1> reboot
Rebooting system. Please stand by ...
```