

Coraid EtherDrive[®]

VS

User's Manual

Draft
Update 8/1/08

Coraid, Inc.
565 Research Drive
Athens, GA 30605
<http://www.coraid.com>

NAME

VS – VirtualStorage Appliance

DESCRIPTION

This introduction documents basic setup and quick start examples for the VS. Please see Appendix A for a description of the VS firmware update procedure. Console connectivity to the VS is currently available by keyboard+video or by serial connection. A serial connection can be established at 9600 baud, 8-N-1.

The VS is a storage virtualization appliance. It allows the storage administrator to flexibly allocate storage capacity as needed, with full redundancy as needed. The VS uses AoE storage as its raw materials, constructing new AoE targets that have the desired size and redundancy characteristics.

The raw storage materials of the VS are 4MB chunks called extents. The extents reside on physical volumes, storage targets prepared for use with the VS. A pool of extents is called a volume group. The VS creates usable storage targets called *LUNs* by allocating extents from a volume group.

A LUN can be mirrored to provide redundancy across multiple physical volumes. For example, an SR has internal redundancy by virtue of its RAID configuration, but it only has one processor. Two SRs are fully redundant, and when the VS mirrors across them, the resulting LUN is resistant not only to a hard disk failure but even to a processor failure in one of the SRs.

This manual documents the commands used to initialize and examine physical volumes, volume groups, LUNs, and mirrors. The most commonly used commands are listed below.

lsoe	display information about AoE targets
lslun	display information about existing LUNs
lspv	display information about physical volumes
lsvg	display information about volume groups
mirror	establish mirroring between two physical volumes
mklegacy	add target to a volume group so that existing data remains accessible
mkln	create a storage target
mkpv	initialize an AoE target for use as a physical volume
mkvg	create a volume group
offline	deactivate a LUN
online	activate a LUN
rmlun	remove a LUN
rmpv	stop using a physical volume
rmvg	remove a volume group
setboot	choose which flash disk area to boot from, A or B
shelf	show or set the VS shelf address
srvshelf	show or set the VS service shelf address
unmirror	break the mirroring relationship between two physical volumes

EXAMPLE**CREATING LUNS**

The following quickstart example sets up the VS with SRs that do not already contain user data.

Set shelf addresses:

```
VS-1:-1> shelf 16
VS16:-1> srvshelf 18
VS16:18>
```

Create volume group:

```
VS16:18> mkvg storage
```

Create physical volumes, which assumes there is a SR on the network with shelf number 9:

```
VS16:18> mkpv 9.6 storage
```

Create the LUN:

```
VS16:18> mklun 0 10g storage
```

Put LUN online:

```
VS16:18> online 0
```

Now all AoE clients will be able to access the storage located at shelf address 18 slot 0, where 18 is the *srvshelf* number and 0 is the LUN number.

CREATING LUNS WITH A LEGACY SR

In order to use existing SR storage on the VS, there are a couple of special steps. These steps must be done to protect data stored on the SR.

Before adding the *legacy* storage, a new physical volume must be available. The reason for the new physical volume is a location for the *metadata* associated with the LUN. The *metadata* is the information the VS uses to maintain state about the storage it is using. Since the *legacy* SRs do not have a place to store that locally, a new physical volume must be added. The new physical volume could be a new *blade* on an existing SR.

Follow these steps to bring the current SR shelf 9 with new SR shelf number 7. First, set the shelf addresses for the VS:

```
VS-1:-1> shelf 16
VS16:-1> srvshelf 18
VS16:18>
```

Create the volume group:

```
VS16:18> mkvg existing
```

Create the physical volume for where *metadata* will be stored:

```
VS16:18> mkpv 7.2 existing
```

Create the LUN:

```
VS16:18> mklegacy 10 9.15 existing
```

Put LUN online:

```
VS16:18> online 10
```

Now all AoE clients will be able to access the storage located at shelf address 18 slot 10, where 18 is the *srvshelf* number and 10 is the LUN number.

SEE ALSO

For more information on the basic concepts of network storage virtualization, see http://en.wikipedia.org/wiki/Storage_virtualization

For a discussion of basic EtherDrive VS concepts with some examples, see the Coraid paper “Understanding the VS: Coraid EtherDrive Storage Virtualization Appliance” at the Coraid website.

NAME

aoestat – show aoe statistics

SYNOPSIS

aoestat

DESCRIPTION

Aoestat displays AoE statistics for the VS unit.

The statistics are presented as running totals and are defined as follows.

msgs – AoE commands received from clients.

wronglun – received AoE commands for LUNs that do not exist or are offline.

wrongshelf – received AoE commands addressed to any shelf other than the AoE serve shelf.

toomanysectors – received AoE commands requesting more data than can fit in a response, or requesting more data than exists on a LUN (I/O off the end of the device).

badflash – received AoE commands that were dropped due to an invalid flash signature.

pvchanging – received AoE commands that were dropped due to an attempted write to a physical extent while it was changing.

intrxlate – received AoE commands that were handled with the fast translation path.

noaarp – received AoE commands that could not be translated to a backing target due to a missing target.

queued – received AoE commands that were queued for processing and could not be handled by the fast translation path.

bcast – received AoE commands that were sent to a broadcast shelf or slot address.

lunoffline – received AoE commands that were queued for processing and sent to an offline LUN.

xprocbadlun – received AoE commands that were queued for processing, but sent to a nonexistent LUN.

wwrongshelf – received AoE commands that were queued for processing, but sent to a shelf address other than the serve shelf.

wrongdiskcmd – received AoE commands that were queued for processing, but sent with an unrecognized AoE ATA command.

wrongaoecmd – received AoE commands that were queued for processing, but sent with an unrecognized AoE command.

opfailed – received AoE commands that were queued for processing, but could not complete due to I/O error.

procxlate – received AoE commands that were queued for processing and successfully processed.

nsplits – received AoE ATA commands that spanned two extents and had to be split into two requests to complete.

resend – AoE commands initiated from the VS that had to be retransmitted.

badtag – received AoE responses that could not be resolved to an outstanding command tag.

cmdfailed – VS initiated AoE commands that failed due to retransmission timeout.

noframes – an Ethernet frame allocation was attempted, but no frames were available to allocate.

NAME

copy - copy one logical-volume to another logical-volume

SYNOPSIS

copy from-LUN to-LUN

DESCRIPTION

Copy initiates a background job to copy the entire data contents from one LUN to another.

From-LUN is the source LUN for the LUN copy. This LUN must already exist. *To-LUN* is the destination LUN for the LUN copy. This LUN must also already exist.

Both LUNs must use the same extent size. The LUN extent size is defined by the volume group from which the LUNs are defined and is viewed with the *lsvg* command.

The *to-LUN* must also be at least as big as the *from-LUN*, or the copy will fail.

The status of the background LUN copy can be viewed with the *wstat* command. A log message is generated when the background copy is initiated.

SEE ALSO

lsvg(1), *pause(1)*, *resume(1)*, *wstat(1)*

NAME

date - show or set date and time

SYNOPSIS

date [*YYYYMMDDhhmm*]

DESCRIPTION

Date is used to print or set the date. If no argument is given, the date will be printed in the following format:

```
Sat Mar 22 12:13:42 GMT 2008
```

To set the date, specify the Year, Month, Day, Hour, and Minute of the time using two digits for each field. It is recommended that the date be set to your local GMT time.

EXAMPLE

```
VS63:63> date 200903221215
Sun Mar 22 12:15:00 GMT 2009
VS63:63> date
Sun Mar 22 12:15:01 GMT 2009
VS63:63>
```

NAME

discover - discover new AoE targets

SYNOPSIS

`discover`

DESCRIPTION

`Discover` sends a broadcast AoE query configuration message onto the network to discover new AoE devices on the network.

The list of discovered AoE devices can be viewed with the *lsaoe* command.

SEE ALSO

lsaoe(1)

NAME

grow, shrink – resize a logical-volume

SYNOPSIS

grow *LUN newsize [physical-volume]*

shrink *LUN newsize*

DESCRIPTION

Grow and *shrink* change the size of the specified *LUN*. Both accept a *newsiz*e argument specifying the total size of the LUN after grow/shrink.

Grow accepts an optional *physical-volume* parameter. When provided, the LUN will be grown using the specified physical volume. The physical volume must reside in the volume group from which the LUN is already allocated.

EXAMPLE

```
VS63:63> mklun 0 500m vg0
2008.07.28 10:54:31 updating 61.0
VS63:63> lslun
LUN          LENGTH  ONLINE  LABEL
0            0.503GB  OFF
VS63:63> grow 0 1g
2008.07.28 10:54:47 updating 61.0
VS63:63> lslun
LUN          LENGTH  ONLINE  LABEL
0            1.002GB  OFF
VS63:63> shrink 0 250m
Shrink lun 0 to 250,000,000 bytes? [n]: y
2008.07.28 10:55:03 updating 61.0
VS63:63> lslun
LUN          LENGTH  ONLINE  LABEL
0            0.251GB  OFF
VS63:63>
```

SEE ALSO

lsvg(1), *mkvg(1)*

NAME

halt - shutdown and wait for key entry to reboot

SYNOPSIS

halt

DESCRIPTION

Halt performs a system shutdown and waits for a key to be pressed. Upon keypress, the *VS* performs a hardware reboot.

NAME

help - print command usage

SYNOPSIS

help [*cmd*]

DESCRIPTION

Help prints the command usage and a short description for a specified *cmd*. Without a *cmd* argument, *help* displays all available commands.

NAME

ifstat - show network interface information

SYNOPSIS

ifstat [*-a*] [*port ...*]

DESCRIPTION

Ifstat shows information about the network interfaces. If no network *port* is specified, all available ports are displayed.

If the *-a* flag is supplied, detailed statistics about the port(s) are displayed.

EXAMPLE

```
VS63:63> ifstat
PORT          ADDR          LINK (Mb/s)
0             0030488f35d6    1000/1000
1             0030488f35d7    1000/1000
VS63:63> ifstat -a 0
PORT          ADDR          LINK (Mb/s)
0             0030488f35d6    1000/1000
Packets Received (64 Bytes): 1418826 129
Packets Received (65-127 Bytes): 94614 24
Packets Received (128-255 Bytes): 587 0
Packets Received (256-511 Bytes): 7194 2
Packets Received (512-1023 Bytes): 117524 25
Packets Received (1024-1522 Bytes): 1121802 72
Good Packets Received: 2760548 253
Broadcast Packets Received: 281308 61
Good Packets Transmitted: 2013533 102
Good Octets Received: 18446744071777760263 0
Good Octets Transmitted: 1981258935 1
Total Octets Received: 18446744071777760263 0
Total Octets Transmitted: 1981258935 1
Total Packets Received: 2760548 253
Total Packets Transmitted: 2013533 102
Packets Transmitted (64 Bytes): 1260897 102
Packets Transmitted (65-127 Bytes): 21 0
Packets Transmitted (1024-1522 Bytes): 752615 0
Broadcast Packets Transmitted: 28194 6
rdfree=155
more...
VS63:63>
```

EXAMPLE CURTAILED FOR BREVITY

NAME

kill - kill background job

SYNOPSIS

kill *workid*

DESCRIPTION

Kill stops the background work specified by *workid*. Work ids are assigned when background work is initiated and can be listed with the *wstat* command.

If a LUN *copy* is terminated, there is no way to restart it at the point it was terminated.

If a physical volume mirror rebuild is terminated, the mirror is broken. This is equivalent to using *unmirror* on the mirror physical volume.

EXAMPLE

```
VS63:63> mirror 61.0 61.1
2008.07.28 12:12:44 updating 61.0
2008.07.28 12:12:45 started rebuilding: 61.0 -> 61.1
VS63:63> wstat
ID          ACTION      DONE          STATE          EXTENT  NP
0          mirror 61.0->61.1  0.07%         rebuilding      91      30
VS63:63> kill 0
2008.07.28 12:13:00 work 0 killed by user
VS63:63> wstat
ID          ACTION      DONE          STATE          EXTENT  NP
0          mirror 61.0->61.1  0.07%         rebuilding (killing) 91      30
VS63:63>
2008.07.28 12:13:06 mirror rebuild killed: 61.0 -> 61.1
2008.07.28 12:13:06 updating 61.0
VS63:63> wstat
VS63:63>
```

SEE ALSO

pause(1), *resume(1)*, *wstat(1)*

NAME

label - set label for LUN

SYNOPSIS

label *logical-volume-number label*

DESCRIPTION

Label associates a textual label with a *logical-volume-number* that will be displayed in the output of the *lslun* command. *Label* is a text string without spaces.

EXAMPLE

```
VS63:63> mklun 0 5g vg0
2008.07.28 12:18:13 updating 61.0
VS63:63> lslun
LUN          LENGTH  ONLINE  LABEL
0            5.003GB  OFF
VS63:63> label 0 accounting
VS63:63> lslun
LUN          LENGTH  ONLINE  LABEL
0            5.003GB  OFF    accounting
VS63:63>
```

SEE ALSO

lslun(1)

NAME

lsaoe – show the status of AoE targets

SYNOPSIS

lsaoe [-c]

DESCRIPTION

lsaoe displays information about the AoE targets on the network. If no arguments are used, *lsaoe* displays the following information for all of the visible AoE targets:

TARGET – the AoE shelf.slot target address

LENGTH – the size of the AoE target in GigaBytes

STATE – the VS communication status with the AoE target

The possible values for this field are:

G – the AoE target responded to an AoE query config and the VS is retrieving the size information from the target

V – the target has completed size and status initialization and is online for use

M – the target was previously visible and valid, but has stopped responding to the VS

PORT – the Ethernet port on which the AoE target is visible

ADDR – the Ethernet address of the AoE target

If a *-c* flag is provided, *lsaoe* displays the AoE config string information stored on the target. The output displays the following information for each AoE target:

TARGET – the AoE shelf.slot target address

LEN – the length of the stored configuration string

CONFIG – the configuration string stored on the target

Note in the example below that unlike the default output, *lsaoe -c* only displays one instance of each target regardless of the PORT it is on.

EXAMPLE

```
VS63:63> lsaoe
TARGET          LENGTH  STATE  PORT          ADDR
8.1             4.294GB  V      0  0060dd475d90
8.1             4.294GB  V      1  0060dd475d90
8.2             4.299GB  V      0  0060dd475d90
8.2             4.299GB  V      1  0060dd475d90
8.3             82.348GB V      0  0060dd475d90
8.3             82.348GB V      1  0060dd475d90
21.1            1000.215GB V      0  0030488b67ca
21.1            1000.215GB V      0  0030488b67cb
21.1            1000.215GB V      1  0030488b67ca
21.1            1000.215GB V      1  0030488b67cb
VS63:63> lsaoe -c
TARGET          LEN  CONFIG
8.1             45  'CoraidPV 77 8.1 -1.-1 0 2 1024 vg0 1 3 8.3.2 '
8.2             0
8.3             46  'CoraidPV 77 8.3 -1.-1 0 2 19633 vg0 1 3 8.3.1 '
21.1            0
VS63:63>
```

NAME

`lslun` - list logical volumes

SYNOPSIS

`lslun [-a] [lun ...]`

DESCRIPTION

`lslun` displays the LUN(s) defined by the VS. One or more LUNs may be specified on the command line. The following information is displayed for each LUN:

LUN - the LUN ID

LENGTH - the size of the LUN in GigaBytes

ONLINE - the online/offline state of the LUN

LABEL - the label assigned to the LUN (if any)

The `-a` flag lists extended information about the LUN including the physical volume(s) the LUN uses and the volume group from which the LUN is defined. Additional debugging information is listed that is useful to Coraid support; undefined information can be ignored.

EXAMPLE

```
VS63:63> lslun
LUN          LENGTH  ONLINE  LABEL
0             5.003GB  OFF     accounting
1             5.003GB  ON
2             5.003GB  OFF
VS63:63> lslun -a 0 1
LUN          LENGTH  ONLINE  LABEL
0             5.003GB  OFF     accounting
           volume group: vg0
           pvs: 61.0
           flag: AoE_Target
           levec=7a031ef0 nle=1194 g=7a1003d0 base=0 limit=0
           cfglen=0 snapshot=0 sslun=0
1             5.003GB  ON
           volume group: vg0
           pvs: 61.0
           flag: AoE_Target online wants_online
           levec=7a02f520 nle=1194 g=7a1003d0 base=0 limit=0
           cfglen=0 snapshot=0 sslun=0
VS63:63>
```

SEE ALSO

label(1)

NAME

`lspv` - list physical volume(s)

SYNOPSIS

`lspv` [*-a*] [*physical-volume*]

DESCRIPTION

`Lspv` displays the physical volumes in all volume groups managed by the VS. One or more PVs may be specified on the command line. The following information is displayed for each PV:

TARGET - the AoE shelf.slot target address

VOLUME GROUP - the volume group to which the PV belongs

LENGTH - the size of the PV in GigaBytes

AVAILABLE - the space available on the PV for VS allocation

NPE - the count of physical extents on the PV

MIRROR - the mirror of the PV, if any

The *-a* flag lists extended information about the PV including LUNs that use the PV. Additional debugging information is listed that is useful to Coraid support; undefined information can be ignored.

EXAMPLE

```
VS63:63> lspv
TARGET          VOLUME GROUP      LENGTH      AVAILABLE      NPE  MIRROR
61.0             vg0               500.103GB   485.075GB     119234
61.1             vg0               500.103GB   500.099GB     119234
61.2             vg0               1000.203GB  1000.198GB    238467
VS63:63> lspv -a 61.0 61.1
TARGET          VOLUME GROUP      LENGTH      AVAILABLE      NPE  MIRROR
61.0             vg0               500.103GB   485.075GB     119234
    luns: 0 1 2
    metadata: 61.0.1
    nvec=119234 data=3582 written = 115652, dirty=115652 cow=0 changing=0 metadata=1 scanned=1
61.1             vg0               500.103GB   500.099GB     119234
    luns:
    metadata: 61.1.1
    nvec=119234 data=0 written = 0, dirty=0 cow=0 changing=0 metadata=1 scanned=1
VS63:63>
```

NAME

`lsvg` - list volume group(s)

SYNOPSIS

`lsvg` [*-a*] [*vg ...*]

DESCRIPTION

`Lsvg` displays the volume groups defined by the VS. One or more LUNs may be specified on the command line. The following information is displayed for each VG:

NAME - the assigned name of the VG

LENGTH - the size of the VG in MegaBytes

AVAILABLE - the space available on the VG for VS allocation

EXTSZ - the extent size of the VG

PVS - the count of PVs currently available / the total count of PVs in the volume group

The *-a* flag lists extended information about the VG including currently available constituent PVs. Additional debugging information is listed that is useful to Coraid support; undefined information can be ignored.

EXAMPLE

```
VS63:63> lsvg
```

NAME	LENGTH	AVAILABLE	EXTSZ	PVS
vg2	1000.203GB	1000.198GB	4MB	01/01
vg1	1000.203GB	1000.198GB	4MB	01/01
vg0	1000.207GB	985.174GB	4MB	02/02

```
VS63:63> lsvg -a vg0 vg1
```

NAME	LENGTH	AVAILABLE	EXTSZ	PVS
vg0	1000.207GB	985.174GB	4MB	02/02
	pvs: 61.0 61.1			
	gen=41 eshift=13 emask=8191 emult=1			
vg1	1000.203GB	1000.198GB	4MB	01/01
	pvs: 61.3			
	gen=1 eshift=13 emask=8191 emult=1			

```
VS63:63>
```

SEE ALSO

`mkvg(1)`

NAME

mirror, unmirror - manage physical-volume redundancy

SYNOPSIS

mirror *physical-volume* *mirror-volume*

unmirror *physical-volume*

DESCRIPTION

Mirror creates a mirror on the *physical-volume* to the *mirror-volume* causing the two volumes to stay in sync. If the *physical-volume* becomes unavailable, the *mirror-volume* will be promoted as the primary and the mirror will be broken.

Unmirror breaks a mirror, releasing the *mirror-volume* for other use.

Upon creation, a rebuild of the mirror will be initiated to copy any dirty extents from *physical-volume* to *mirror-volume*. A log message is generated to indicate the rebuild has started. The status of the rebuild can be viewed with the *wstat* command.

SEE ALSO

mkpv(1), *pause(1)*, *resume(1)*, *wstat(1)*

NAME

mklegacy - legacy create a volume group

SYNOPSIS

mklegacy *lun-number* *aoe-target* *volume-group*

DESCRIPTION

Mklegacy creates LUNs comprised of SR storage that already contains user data permitting the use of legacy SR storage with the VS.

Lun-number is the LUN used to export the legacy volume. *Aoe-target* is the shelf.slot target of the legacy physical volume. The physical volume will be imported into the specified *volume-group*.

The *volume-group* must already exist and contain at least one physical volume with an allocatable extent for the legacy physical volume's metadata. Normally the metadata for a physical volume is stored on the physical volume itself, but this is not possible with legacy physical volumes. For a full explanation of the legacy physical volume support, please see the document *VS Principles of Operation*.

EXAMPLE

A sample of setting up a legacy LUN:

```
VS63:63> lspv
TARGET          VOLUME GROUP          LENGTH          AVAILABLE          NPE  MIRROR
61.0             vg0                    500.103GB      500.099GB          119234
VS63:63> mklegacy 0 61.1 vg0
2008.07.28 14:22:32 updating 61.1
2008.07.28 14:22:32 updating 61.0
VS63:63> lslun -a 0
LUN             LENGTH  ONLINE  LABEL
0               500.103GB  OFF
    volume group: vg0
    pvs: 61.0 61.1
    flag: AoE_Target
    levec=7a017100 nle=119235 g=7a1003d0 base=0 limit=0
    cfglen=0 snapshot=0 sslun=0
VS63:63>
```

In this example, the legacy LUN is being created for physical volume *61.1*. The metadata for the legacy LUN is stored on physical volume *61.0*.

BUGS

Due to the current implementation of the VS legacy feature some data at the end of the legacy device may be ignored. The VS addresses physical volumes in units of extents. The extent size is defined by the extent size of the volume group the physical volume resides in. When importing a pv into a volume group, the size of the pv is divided into extentsize units; up to extentsize - 1 remainder bytes at the end of the device are ignored. This is a problem for legacy physical volumes as these ignored bytes may contain user data.

To workaround this problem users should ensure they are not using this extentsize - 1 remainder bytes on a legacy AoE target before making it a legacy physical volume. The simplest method for doing this is to shrink the use of the AoE target by extentsize bytes. The method for doing this will vary based on the use of the AoE target. Consult your filesystem/database documentation to see if a shrink utility is available.

SEE ALSO

lsaoe(1), *lsvg(1)*, *mkpv(1)*, *mkvg(1)*

NAME

mklun - create a logical volume

SYNOPSIS

mklun *lun-number size volume-group* [*physical-volume*]

mklun *lun-number flash* [A|B]

DESCRIPTION

Mklun creates a logical volume backed by AoE storage or by flash boot images for updating the appliance.

Lun-number is the LUN to use in exporting the data. The valid range for *lun-number* is 0 - 254.

Size is the size of the logical volume being created. In specifying the size, the following suffixes can be used to indicate the size.

m, M - Megabyte

g, G - Gigabyte

t, T - Terabyte

If no suffix is specified the size is assumed to be in Megabytes. *Volume-group* is the volume group that has already been created using the *mkvg* command.

By default, the LUN data is allocated from the physical volumes in *Volume-group* in an arbitrary order. If a *Physical-volume* argument is supplied, the LUN will be allocated from the specified physical volume. The physical volume must reside in the defined *Volume-group*.

A *flash logical-volume* can be created to upgrade the appliance firmware. There are two firmware images stored on the flash, *A* and *B*.

EXAMPLE

```
mklun 0 100M vg1
```

To create a logical volume of 100 Megabytes using volume-group *vg1*.

```
mklun 10 20t vg7
```

To create a logical volume of 20 Terabytes using volume-group *vg7*.

```
mklun 6 flash A
```

To create logical volume number 6. This logical volume can be used to copy a different software image onto the VS.

DIAGNOSTICS

If the PLD (Per LUN Data) is not able to be written to the physical-volume, the following log message is displayed:

```
pldsave failed: shelf.slot error
```

Shelf.slot defines the physical volume target and *error* defines the error that occurred.

The LUN will be created, but will not survive reboot.

SEE ALSO

lslun(1), *mklegacy(1)*, *mkpv(1)*, *offline(1)*, *shelf(1)*

NAME

mkpv – add a physical volume to a volume group

SYNOPSIS

mkpv target volume-group

DESCRIPTION

Mkpv makes an AoE target a physical volume, adding it to a volume group.

DIAGNOSTICS

The *target* must not already be in use.

The *volume-group* must already exist, having been created with *mkvg*.

The *target* must be listed as available in the output of *lsaoe*.

SEE ALSO

lspv(1), *lsaoe(1)*, *mkvg(1)*

NAME

`mkvg` - make a volume group

SYNOPSIS

`mkvg volume-group`

DESCRIPTION

Mkvg creates an empty volume group.

Volume-group is the string used to name the new volume group. The name of the volume group is limited to 32 characters.

EXAMPLE

```
mkvg vg-test
```

To create the volume-group *vg-test* using the default extent-size of 4 Megabytes.

DIAGNOSTICS

If a serve shelf has not been set with the *srvshelf* command, the *mkvg* command will fail.

BUGS

Only 4MB extents are currently supported.

SEE ALSO

shelf(1), *lsvg(1)*

NAME

model - display VS hardware model

SYNOPSIS

model

DESCRIPTION

Model displays the VS hardware model.

EXAMPLE

```
VS63:63> model
VS21
VS63:63>
```


NAME

offline, online - change status of a LUN

SYNOPSIS

offline logical-unit-number

online logical-unit-number

DESCRIPTION

Offline makes the specified *lun-unit-number* unavailable for use.

Online makes the specified *lun-unit-number* available for use.

NAME

passwd - change the password

SYNOPSIS

passwd

DESCRIPTION

The *passwd* command is used to set the VS login password.

If necessary, *passwd* first prompts for the current password. Upon a correct match of the current password, it prompts for the new password twice. Upon a match in the new password, the VS password is successfully changed.

The password is limited to 32 characters. Setting an empty password clears the password.

If you forget your password, enter the password *ivelostit*, and email support@coraid.com the challenge output. The support team will send the response to the challenge. Entering a successful response to the password reset challenge clears the password. The password *ivelostit* is reserved and may not be used.

EXAMPLE

```
VS63:63> passwd
  new password: ****
again to verify: ****
password changed
VS63:63> exit
Login[63:63]: ****
VS63:63> passwd
old password: ****
  new password:
again to verify:
password changed
VS63:63> exit
Console password unset.  Access granted.
VS63:63>
```

NAME

pause - pause background work in progress

SYNOPSIS

pause *workid*

DESCRIPTION

Pause temporarily stops background work in progress. The background work can be restarted using the *resume* command. The status of all background work is visible via the *wstat* command.

EXAMPLE

```
VS63:63> mirror 61.0 61.2
2008.07.28 14:48:39 updating 61.0
2008.07.28 14:48:39 updating 61.1
2008.07.28 14:48:40 started rebuilding: 61.0 -> 61.2
VS63:63> wstat
ID          ACTION      DONE          STATE          EXTENT  NP
0          mirror 61.0->61.2  0.02%         rebuilding     3612   30
VS63:63> pause 0
2008.07.28 14:48:44 work id 0 paused
VS63:63> wstat
ID          ACTION      DONE          STATE          EXTENT  NP
0          mirror 61.0->61.2  0.02%         rebuilding (pausing)  3612   28
VS63:63> wstat
ID          ACTION      DONE          STATE          EXTENT  NP
0          mirror 61.0->61.2  0.02%         rebuilding (paused)  3612   0
VS63:63>
```

BUGS

Pause simply directs the background work to pause itself when it can. The only reliable way to know the work is fully paused is to verify the *wstat* output displays the work as (*paused*).

SEE ALSO

kill(1), *mirror(1)*, *resume(1)*, *wstat(1)*

NAME

ps - show processes

SYNOPSIS

ps [-m]

DESCRIPTION

Ps displays information about the processes currently running in the system. Without any arguments, the output is formatted as follows:

TIME	STATE	NAME
------	-------	------

TIME is the amount of run time that process has had since the last reboot.

STATE is the current state of the process, which can be *ready*, *running*, or *sleeping*.

NAME is the name of the process.

If the optional *-m* argument is used, the output is formatted as follows:

NAME	WCHAN	CALLERPC	AVAIL/STACK	STATE
------	-------	----------	-------------	-------

NAME is the name of the process.

WCHAN is information about an internal communication channel.

CALLERPC is information about the internal program counter.

AVAIL/STACK is information about the stack size being used.

STATE is the current state of the process, which can be *ready*, *running*, or *sleeping*.

EXAMPLE

```
VS> ps
```

TIME	STATE	NAME
0:00.03	running	main
0:00.16	sleeping	builder
0:00.64	sleeping	worker
0:00.00	sleeping	worker
0:00.00	sleeping	worker
0:00.00	sleeping	worker
0:00.00	sleeping	worker

```
VS> ps -m
```

NAME	WCHAN	CALLERPC	AVAIL/STACK	STATE
main	: 0003293c	0000fb27	40,900/131,072	running
builder	: 0003296c	000132b9	130,980/131,072	sleeping
worker	: 7d501280	000226a6	131,008/131,072	sleeping
worker	: 7d501280	000226a6	131,008/131,072	sleeping
worker	: 7d501280	000226a6	131,008/131,072	sleeping
worker	: 7d501280	000226a6	131,008/131,072	sleeping

NAME

pstat - show process debugging information

SYNOPSIS

pstat

DESCRIPTION

Pstat prints statistics about the processes in the system, which includes the percent idle cycles, the number of ready processes, and the number of sleeping processes.

EXAMPLE

```
VS63:63> pstat  
idle 100%, 0 ready, 96 sleeping
```

NAME

reboot - shutdown and restart VS

SYNOPSIS

reboot

DESCRIPTION

Reboot performs a system shutdown and hardware reset causing the VS to immediately reboot.

NAME

release - show software release date

SYNOPSIS

release displays the release date of the currently running version of firmware.

EXAMPLE

```
VS> release  
Running Release: Fri Jan 18 13:00:46 EST 2008
```

NAME

resume - continue paused work

SYNOPSIS

resume *workid*

DESCRIPTION

Resume restarts background work that has previously been paused using the *pause* command. The status of all background work is visible via the *wstat* command.

SEE ALSO

kill(1), *mirror(1)*, *pause(1)*, *wstat(1)*

NAME

rmlun - remove logical volume

SYNOPSIS

rmlun *logical-volume-number*

DESCRIPTION

Rmlun removes the logical volume specified by *logical-volume-number*. A logical volume must be *offline* to be removed.

A confirmation prompt is displayed to confirm the requested action. Once a logical volume is removed, it cannot be rebuilt. All data associated with it is lost.

SEE ALSO

offline(1), *lslun(1)*

NAME

rmpv - remove physical volume

SYNOPSIS

rmpv *physical-volume*

DESCRIPTION

Rmpv removes the specified *physical volume* from its volume group. A physical volume cannot be removed if it is used by any LUN.

SEE ALSO

lspv(1)

NAME

rmvg - remove volume group

SYNOPSIS

rmvg *volume-group*

DESCRIPTION

Rmvg removes the *volume-group* from the system. The *volume-group* must not contain any physical volumes.

SEE ALSO

lsvg(1)

NAME

setboot - set image to boot from

SYNOPSIS

```
setboot [ A | B ]
```

DESCRIPTION

Setboot sets the default boot image to one of *A* or *B*. If no argument is given, it will print the current setting.

EXAMPLE

```
VS63:63> setboot
will boot from A
VS63:63> setboot B
VS63:63> setboot
will boot from B
VS63:63>
```

BUGS

It is not currently possible to display the release of images A and B from the command line.

NAME

sethash, showhash, verifyhash – set, view, or verify flash signature

SYNOPSIS

```
sethash sig_s sig_r
```

```
showhash
```

```
verifyhash
```

DESCRIPTION

Every VS has a unique signature that validates it as a Coraid product. The VS must have a valid signature to pass AoE traffic. As each signature is unique to the VS on which it is installed, a signature **cannot** be copied from one unit to another.

Sethash sets the hash signature in the VS flash. This command should only be run at the direction of Coraid's support staff as it **can destroy a valid signature**.

Showhash shows the hash key input used to generate the hash signature.

Verifyhash verifies the hash signature stored in flash.

EXAMPLE

```
VS> verifyhash  
hash key is good  
VS> verifyhash  
hash key is bad
```

NAME

shelf, srvshelf – display or set administrative and AoE shelf addresses

SYNOPSIS

```
shelf [ shelf-number ]
```

```
srvshelf [ shelf-number ]
```

DESCRIPTION

Shelf displays the current administrative (admin) shelf number. When provided with a *shelf-number* argument the admin shelf number is set to *shelf-number*.

The admin shelf address is used for management of the VS. It is used for flash update LUNs and CEC communication.

Srvshelf displays the current AoE serve shelf number. When provided with a *shelf-number* argument the serve shelf number is set to *shelf-number*.

The AoE serve shelf address is used to export AoE (SR) backed storage LUNs.

The admin and AoE serve shelf addresses may be set to any value between 0 and 65534.

EXAMPLE

```
VS63:63> shelf
shelf=63  srvshelf=63
VS63:63> shelf 500
VS500:63> shelf
shelf=500  srvshelf=63
VS500:63>
```

BUGS

CEC functionality is not currently available.

SEE ALSO

mklun(1)

NAME

sos - output information for support

SYNOPSIS

sos

DESCRIPTION

Sos displays information needed by Coraid support for supporting the VS appliance. Please capture the output in its entirety and send it along with any questions when contacting support about VS issues.

TEMP(1)

TEMP(1)

NAME

temp - display CPU temperature

SYNOPSIS

temp

DESCRIPTION

Temp displays the temperature of the CPU in degrees celcius.

NAME

wstat - list background work

SYNOPSIS

wstat

DESCRIPTION

The VS performs background work on the user's behalf for LUN copying and physical volume mirroring. *Wstat* displays the status of the background work in process. The output is formatted as follows:

ID - the numerical id of the background work entity

ACTION - a synopsis of the work being performed

DONE - the percent of total work completed

STATE - the work state

Work that has been paused or killed with the *pause* or *kill* commands will be listed as (*paused*), (*pausing*), or (*killing*) depending on whether all background processes performing the work have acknowledged the pause/kill request.

EXTENT - the highest numbered extent being processed

NP - the number of worker processes currently performing the work

EXAMPLE

```
VS63:63> wstat
```

ID	ACTION	DONE	STATE	EXTENT	NP
0	mirror 61.0->61.2	0.05%	rebuilding	3642	30

```
VS63:63>
```

SEE ALSO

kill(1), *pause(1)*, *resume(1)*

NAME

Appendix A - Updating VS Firmware

DESCRIPTION

The VS firmware update procedure is very similar to the firmware update procedure for the SR. A special update LUN is created that exports the flash boot image to be updated. From the initiator, the new firmware release is written to the update LUN using coraid-update or dd. Unlike the SR, there is no "update" command to run on the VS after writing the new firmware image to the LUN from the client; a simple reboot is all that's necessary to run the updated firmware.

EXAMPLE

```
VS63:63> release
Running Release: Tue Jul 29 14:45:40 EDT 2008
VS63:63> setboot
will boot from A
VS63:63> mklun 15 flash A
VS63:63> online 15
2008.07.30 12:30:50 putting lun 15 online
VS63:63> lslun 15
LUN          LENGTH  ONLINE  LABEL
15           0.008GB   ON      flash area A
VS63:63>
```

On the client system:

```
% aoe-stat | grep 63
   e63.15          0.008GB          eth0 8192  up
% # Verify the md5sum with that reported by the VS support page
% md5sum VS20080731
945be5f330fdb7944c10673842a8bc5b  VS20080731
% coraid-update VS20080731 /dev/etherd/e63.15
% # Or alternately, if your system lacks coraid-update
% dd if=VS20080731 of=/dev/etherd/e63.15
291+1 records in
291+1 records out
149060 bytes (149 kB) copied, 0.219686 s, 679 kB/s
%
```

Back on the VS appliance:

```
VS63:63> reboot

rebooting...

Coraid Boot ROM

Boot A?
booting...
Done
2,145,779,712 bytes of memory

VS: CORAID EtherDrive Storage Virtualization Appliance

VS Release: Thu Jul 31 12:38:28 EDT 2008
BIOS build 11/02/07
2,145,779,712 bytes
Flash size 32 MB
maxpe=68,572,963
Console password unset.  Access granted.
VS63:63>
```


The fonts used in this manual were Lucida Sans, in a special version incorporating over 1700 characters from the Unicode Standard, along with Lucida Sans Italic, Lucida Sans DemiBold, and Lucida Typewriter, designed by Bigelow & Holmes, Atherton, California. The hinted Adobe Type 1 representation of the fonts was provided by Y&Y Inc., 45 Walden Street, Concord, MA, 01742, USA.