Chapter 17
-
VERITAS Volume Manager (VxVM)

INDEX

Introduction

VERITAS Volume Manager 3.5 for HP-UX provides state-of-the-art online disk management for HP-UX. VxVM consists of several related products:

Base VERITAS Volume Manager 3.5 for HP-UX (Base VxVM) Base-VXVM:

provides basic volume manager features, including a Java-based GUI, and is included with the HP-UX 11i (Internet), Enterprise, and Mission Critical Operating Environments, as well as with the HP-UX 11i Application Release for no additional fee. When MC/ServiceGuard or ServiceGuard OPS Edition is installed on your system, the Base VERITAS Volume Manager 3.5 for HP-UX also provides some basic clustering features.

VERITAS Volume Manager 3.5 for HP-UX (VxVM) B9116AA:

provides a full set of enhanced volume manager capabilities, including mirroring, and is available for an additional fee.

VERITAS Cluster Volume Manager 3.5 for HP-UX (CVM) B9117AA:

provides enhanced volume manager functionality for clustered environments, is integrated with MC/ServiceGuard and ServiceGuard OPS Edition, and is available for an additional fee. B9117AA requires B9116AA.

VERITAS Volume Manager 3.5 FastResync Option for HP-UX (FR) B9118AA:

reduces the time taken to resynchronize a split mirror to the volume, and is available for an additional fee. B9118AA requires B9116AA.

Feature Availability

VERITAS Volume Manager 3.5 for HP-UX Feature Availability

Feature	AR 0902 Base VERITAS Volume Manager 3.5 for HP-UX (Base-VXVM)	AR 0902 VERITAS Volume Manager 3.5 for HP-UX (B9116AA)	AR 0902 VERITAS Cluster Volume Manager 3.5 for HP-UX (B9117AA)	AR0902 VERITAS Volume Manager 3.5 FastResync Option for HP-UX (B9118AA)	VERITAS Volume Manager 3.5 for HP-UX
Java-based admin GUI	Supported	Supported	Supported	-	Supported
Striping (RAID 0)	Supported	Supported	Supported	-	Supported
Concatenation	Supported	Supported	Supported	-	Supported
Path failover support (active/passive peripherals)	Supported	Supported	Supported	-	Supported

Online resizing of volumes	Supported	Supported	Supported	-	Supported
Load balancing – DMP (active/active peripherals)	-	Supported	Supported	-	Supported
Hot-relocation and unrelocation	-	Supported	Supported	-	Supported
Mirroring (RAID-1)	-	Supported	Supported	-	Supported
Number of mirrors supported	-	32	32	-	Supported
Mirrored Stripes (RAID 0+1)	-	Supported	Supported	-	Supported
Striped Mirrors (RAID 1+0)	-	Supported	Supported	-	Supported
RAID-5	-	Supported	-	-	Supported
Online migration	-	Supported	Supported	-	Supported
Online relayout	-	Supported	Supported	-	Supported
Task monitor	Supported for base VXVM but not for CVM	Supported	Not supported	-	Supported
FastResync		-	-	Yes	Supported
Support for MC/SG (A.11.13 and A.11.14)	Supported	Supported	Supported		With additional license
Support for SG OPS Edition	Supported for CVM but not for base VxVM	-	Supported	-	Supported for CVM but not for base VxVM
Multiple node cluster support with VxVM	16 MC/SG	16 MC/SG	-	-	16 MC/SG
Multiple node cluster support with CVM	4 MC/SG2 SG OPS	-	4 MC/SG4 SG OPS	-	4 MC/SG2 SG OPS
Online Reconfiguration for shared disk groups	Disk group can be activated on only 1 node	-	Disk group can be activated on up to 4 nodes	-	Disk group can be activated on only 1 node

swlist Output and Licencing

```
# swlist –l product Base-VXVM B9116AA
# Initializing...
# Contacting target "grcdg119"...
#
# Target:  grcdg119:/
#
# B9116AA              3.5m        VERITAS Volume Manager 3.5 for HP-UX
  B9116AA.VxVM-LIC     3.5m        VERITAS Volume Manager 3.5 for HP-UX
# Base-VXVM            B.03.50.5   Base VERITAS Volume Manager Bundle 3.5 for HP-UX
  Base-VXVM.VRTSvxvm   3.5m        Base VERITAS Volume Manager 3.5 for HP-UX
  Base-VXVM.VRTSvmdoc  3.5m        VERITAS Volume Manager Documentation
  Base-VXVM.VRTSfspro  3.5-ga08    VERITAS File System Management Services Provider
  Base-VXVM.VRTSvmpro  3.5m        VERITAS Volume Manager Management Services
                                   Provider
  Base-VXVM.VRTSobgui  3.0.2.261a  VERITAS Enterprise Administrator
  Base-VXVM.VRTSob     3.0.2.261a  VERITAS Enterprise Administrator Service
  Base-VXVM.VRTSvlic   3.00.007e   VERITAS License Utilities
```

Confirm that the Product is Licensed

```
# vxlicense -t VxVM

vrts:vxlicense: INFO: Feature name: VxVM [95]
vrts:vxlicense: INFO: Number of licenses: 1 (non-floating)
vrts:vxlicense: INFO: Expiration date: Sun Mar 14 09:00:00 2004 (340.9 days from
now)
vrts:vxlicense: INFO: Release Level: 31
vrts:vxlicense: INFO: Machine Class: All

# vxlicense -p (shows all licenses)

vrts:vxlicense: INFO: Feature name: VxVM [95]
vrts:vxlicense: INFO: Number of licenses: 1 (non-floating)
vrts:vxlicense: INFO: Expiration date: Sun Mar 14 09:00:00 2004 (340.9 days from
now)
vrts:vxlicense: INFO: Release Level: 31
vrts:vxlicense: INFO: Machine Class: All

vrts:vxlicense: INFO: Feature name: CVM [68]
vrts:vxlicense: INFO: Number of licenses: 1 (non-floating)
vrts:vxlicense: INFO: Expiration date: Sun Mar 14 09:00:00 2004 (340.9 days from
now)
vrts:vxlicense: INFO: Release Level: 31
vrts:vxlicense: INFO: Machine Class: All

vrts:vxlicense: INFO: Feature name: HP_OnlineJFS [50]
vrts:vxlicense: INFO: Number of licenses: 1 (non-floating)
vrts:vxlicense: INFO: Expiration date: Sun Aug 24 10:00:00 2003 (137.9 days from
now)
vrts:vxlicense: INFO: Release Level: 35
vrts:vxlicense: INFO: Machine Class: All
```

see also

```
# ll /etc/vx/elm/*.lic
-rw-r--r--   1 root       sys            170 Apr  8 12:59 /etc/vx/elm/50.lic
-rw-r--r--   1 root       sys            170 Apr  8 12:58 /etc/vx/elm/68.lic
-rw-r--r--   1 root       sys            170 Apr  8 12:58 /etc/vx/elm/95.lic
```

Architecture

Init scripts - VxVM 3.5

via /etc/inittab

- /sbin/init.d/vxvm-sysboot
  - Starts DMP restore daemon (/dev/vx/dmpconfig)
  - Starts configuration daemon in boot mode (/dev/vx/config)
  - Reads /etc/vx/volboot file to determine disk ownership for rootdg

- /sbin/init.d/vxvm-startup
  - Will not start if /etc/vx/reconfig.d/state.d/install-db exist
  - Starts up IO daemons
  - Creates disk acces records for all devices
  - Imports disk groups
  - Start configuration daemon in enabled mode
  - Creates DMP nodes that the kernel has detected
  - Starts volumes

Runlevel 0

- /sbin/rc0.d/K930vxvm-daemon-kill -> /sbin/init.d/vxvm-daemon-kill
  - Execute /usr/sbin/vxiod to do away with all the Volume Manager I/O daemon

Runlevel 1

- /sbin/rc1.d/S091vxvm-nodes-check -> /sbin/init.d/vxvm-nodes-check
  - Creates special nodes /dev/vx/
- /sbin/rc1.d/S092vxvm-startup -> /sbin/init.d/vxvm-startup
  - Will not start if /etc/vx/reconfig.d/state.d/install-db exist
  - Starts up IO daemons
  - Creates disk acces records for all devices
  - Imports disk groups
  - Start configuration daemon in enabled mode
  - Creates DMP nodes that the kernel has detected
  - Starts volumes
- /sbin/rc1.d/S093vxvm-reconfig -> /sbin/init.d/vxvm-reconfig
  - Performs VxVM reconfiguration (if needed)
  - Called by vxinstall to add new disks
  - On systems with LVM called by conversion utilities
- /sbin/rc1.d/K999isisd -> /sbin/init.d/isisd
  - Stops the Isis server daemon

Runlevel 2

- /sbin/rc2.d/S096vxvm-recover -> /sbin/init.d/vxvm-recover
  - Performs volume recovery and resynchronization

  - Starts relocation daemon
- /sbin/rc2.d/S994vxnm-vxnetd -> /sbin/init.d/vxnm-vxnetd
  - Checkes if VVR (Veritas Volume Replicator) license is installed
  - Starts vxnetd if VVR license is installed
- /sbin/rc2.d/S996vradmind -> /sbin/init.d/vras-vradmind.sh
  - Starts vradmind daemon (Volume Replicator Administrative Services) only if VVR license is installed
- /sbin/rc2.d/S996vxrsyncd -> /sbin/init.d/vxrsyncd.sh
  - Starts in.vxrsyncd daemon only if VVR license is installed
- /sbin/rc2.d/S999isisd -> /sbin/init.d/isisd
  - Starting VERITAS Enterprise Administrator Service (vxsvc)

vxconfigd – vxdctl

The VERITAS Volume Manager (VxVM) configuration daemon, **vxconfigd**, maintains disk configurations and disk groups in VxVM. **vxconfigd** takes requests from other utilities -- such as **vxassist** or the VEA Gui -- for configuration changes, and communicates those changes to the kernel and modifies configuration information stored on disk. **vxconfigd** also initializes VxVM when the system is booted.
The **vxdctl** utility manages aspects of the state of the volume configuration daemon, **vxconfigd**, and also manages aspects of configuration for bootstrapping the rootdg disk group.
The **volboot** file (*/etc/vx/volboot*) forms a key element in managing the state of vxconfigd and bootstrapping the rootdg disk group. This file contains a host ID that VERITAS Volume Manager (VxVM) uses to establish ownership of physical disks. The host ID is used to ensure that two or more hosts do not interfere with each other when accessing disks on a shared SCSI bus or Storage Area Network (SAN). VxVM also uses host IDs to generate unique ID strings for stamping disks and disk groups.

vxiod

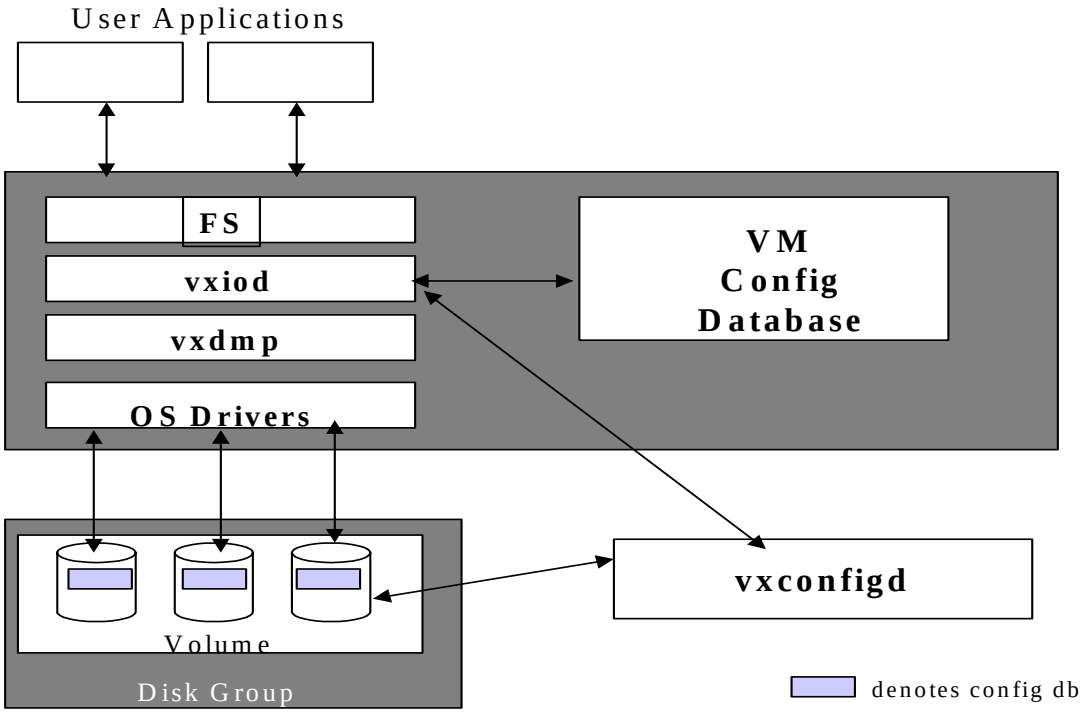
The **vxiod** utility starts, stops, or reports on VERITAS Volume Manager (VxVM) I/O daemons. An I/O daemon provides a process context for performing I/O in VxVM. VERITAS Volume Manager I/O daemons are not required for correct operation, but not having I/O daemons can adversely affect system performance.

The number of daemons to create for general I/O handling is dependent on system load and usage. If volume recovery seems to proceed slower at times, it may be worthwhile to create more daemons.

Each I/O daemon starts in the background and creates an asynchronously-running process, which detaches itself from the controlling terminal and becomes a volume I/O daemon. The **vxiod** utility does not wait for these processes to complete.

vxrelocd (hotrelocation / unrelocation); vxnotify

The **vxrelocd** command monitors VERITAS Volume Manager (VxVM) by analyzing the output of the vxnotify command, and waits for a failure. When a failure occurs, **vxrelocd** sends mail via mailx to root (by default) or to other specified users and relocates failed subdisks. After completing the relocation, **vxrelocd** sends more mail indicating the status of each subdisk replacement. The **vxrecover** utility is then run on volumes with relocated subdisks to restore data. Mail is sent after **vxrecover** executes.



VxVM Objects

Disk Groups (e.g. roodg)

A collection of VM–Disks. A disk group has one ore more volumes.
Disk drives can be shared by two or more hosts, but accessed by only one host at a time. If one hosts crashes, the other host can take over its disk groups and therefore, its disks.

Volumes

A volume with a plex only belongs to one disk group.

Plexes (plex01, plex02)

If you have 2 plexes, you have 1 mirror. Plexes can also be called mirrors. A plex consists of one or more subdisks.

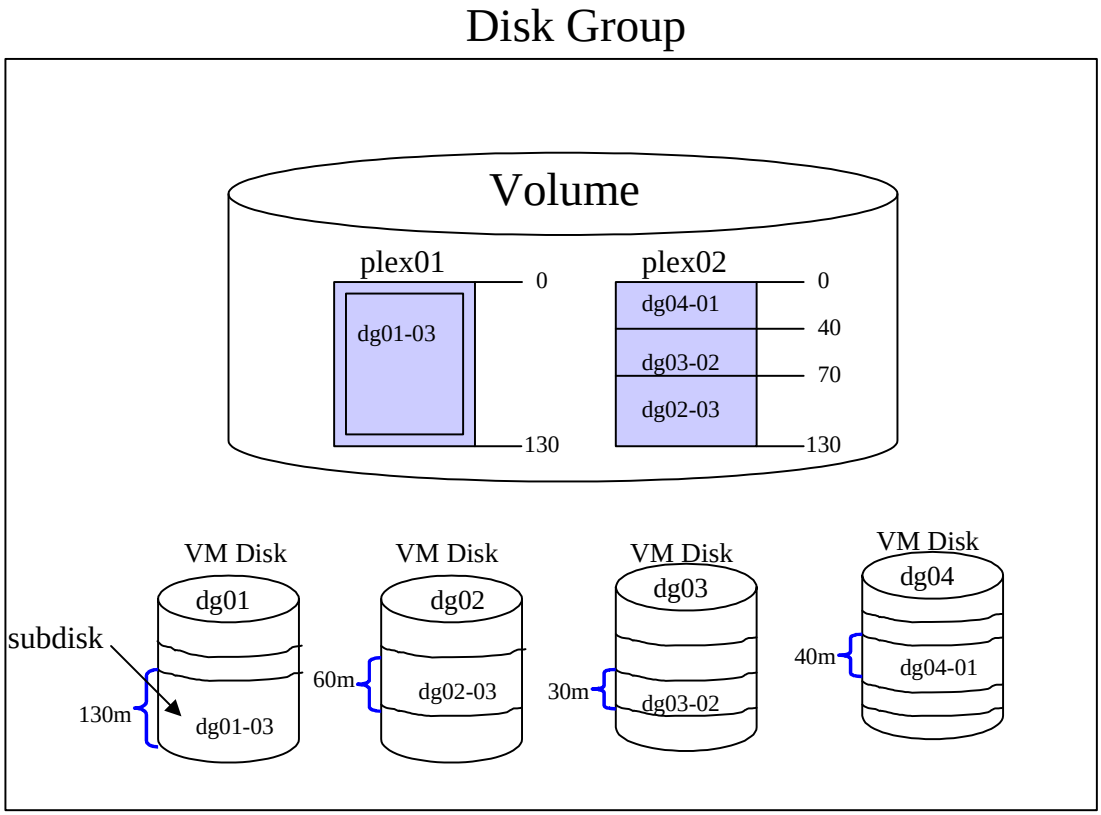
Volume Manager Disks (VM Disk)

VM-Disks are physical disk that are under VxVM control and contain one or more subdisks. The VM-Disks consist of private and public region:

private region	public region
1. disk header – label – diskgroup info: – disk media name (disk-0X) – diskid – disk access name (c#t#d#) – hostid – pointers to the private and public regions `vxdisk list <diskname>` 2. configuration database – contains the VxVM objects 3. kernel logs – contains configuration changes (log plexes attached, object creation / deletion)	The public region is used for allocating sudisk to use it for: data storage

Subdisks (dg0x-0y)

Volume's subdisks and plexes are stored in the same disk group. The number of subdisks on a VM Disk is limited only to the number of VxVM object records the private region can hold.



VMSA - Volume Manager Storage Administrator (VxVM 3.2)

Confirm that the vmsa server is not running

```
# vmsa_server -q

Volume Manager Storage Administrator Server: not running
```

Start the vmsa server

```
# vmsa_server &    or    # vmsa_server -s

# vmsa_server –q
Volume Manager Storage Administrator Server: running
```

Stop the vmsa server

```
# vmsa_server -k

# vmsa_server –q
Volume Manager Storage Administrator Server: not running
```

Auto-Start Activation

```
Auto-start mode starts whenever a VMSA client retries to connect to the
server.

# ./opt/HPvmsa/bin/autostart on /opt/HPvmsa
Adding server auto-start lines to /etc/services...
Adding server auto-start lines to /etc/inetd.conf...
Removing boot-time server start up script...
HANGUP inetd
```

Confirm that Auto-Start is enabled

```
# netstat -a | grep vmsa
tcp        0      0  *.vmsa             *.*                LISTEN
```

VMSA directories

```
# ll /var/opt/vmsa/logs/                    logfiles
-rw-r--r--   1 root       root             18 May  9 10:38 .server_pids
-rw-r--r--   1 root       sys            3514 May  9 10:38 access
-rw-r--r--   1 root       sys            1128 May  9 10:57 command
-rw-r--r--   1 root       sys            5872 May  9 10:38 server.log

# ll /opt/HPvmsa/bin/                       commands for start/stop
-rwxr-xr-x   1 bin        bin            6308 Sep 18  2000 autostart
-rwxr-xr-x   1 bin        bin            2562 Sep 18  2000 vmsa
-rwxr-xr-x   1 bin        bin            7635 Sep 18  2000 vmsa_server

# vi /opt/HPvmsa/vmsa/properties          customization of the GUI properties
```

VEA - Veritas Enterprise Administrator (VxVM 3.5)

You use the VxVM VEA to administer disks, volumes, and file systems on local or remote machines. VxVM VEA is a Java-based interface that consists of a server (vxsvc) and a client (vea). The server runs on a UNIX machine that is running VxVM. The client runs on any machine that supports the Java Runtime Environment.

Confirm that the vxsvc server is not running

```
# /opt/VRTSob/bin/vxsvc –m
Current state of server : NOT RUNNING
```

Start the vxsvc

```
# /opt/VRTSob/bin/vxsvc
# /opt/VRTSob/bin/vxsvc –m
Current state of server : START PENDING          During startup
# /opt/VRTSob/bin/vxsvc –m
Current state of server : RUNNING                vxsvc is up and running
```

Comment: /sbin/rc2.d/S999isisd starts vxsvc

Stop a running vxsvc

```
# /opt/VRTSob/bin/vxsvc –k
Request is in process..
VEA Server is shutting down  .
Server was shutdown successfully
```

Start the client on a UX 11i system

```
# /opt/VRTSob/bin/vea
```

Additional information about the server or the client you will find in the README files located in /opt/VRTSob/ or vxsvc(1M) and vea(1M). The User's Guide – Veritas Enterprise Administrator (vxvm_ug.pdf) can be found in /usr/share/doc/vxvm.

Log & Lock files

```
# ll /var/vx/isis/
total 64
-rw-------   1 root       sys           1024 Apr  8 16:21 .rnd
drwxr-xr-x   3 root       sys             96 Oct  1  2002 alertlog
-rw-r--r--   1 root       sys              0 Oct  1  2002 command.log
-rwxrwxrwx   1 root       root            16 Apr  8 16:22 state
drwxr-xr-x   2 root       sys             96 Oct  1  2002 tasklog
----------   1 root       sys              5 Apr  8 16:21 vxisis.lock
-rw-r--r--   1 root       root           839 Apr  8 16:27 vxisis.log
```

Disk Group Tasks

Diskgroup specifical listings

```
# vxdg free
GROUP        DISK         DEVICE       TAG          OFFSET    LENGTH    FLAGS
rootdg       rootdisk01   c3t15d0      c3t15d0      8192000   9586431   -
rootdg       rootdisk02   c1t15d0      c1t15d0      8192000   9586431   -
ccdg         ccdg01       c4t0d1       c4t0d1       0         25163136  -

# vxdg list
NAME         STATE           ID
rootdg       enabled  1042879605.1025.hprtlc35
ccdg         enabled  1042885455.1195.hprtlc35

# vxdg list rootdg
Group:     rootdg
dgid:      1042879605.1025.hprtlc35
import-id: 0.1
flags:
version:   90
local-activation: read-write
detach-policy: global
copies:    nconfig=default nlog=default
config:    seqno=0.1196 permlen=727 free=715 templen=5 loglen=110
config disk c1t15d0 copy 1 len=727 state=clean online
config disk c3t15d0 copy 1 len=727 state=clean online
log disk c1t15d0 copy 1 len=110
log disk c3t15d0 copy 1 len=110
```

Adding a Disk to VxVM

`/etc/vx/bin/vxdisksetup` creates private and public regions on the disk, but does not add the disk to a disk group

Initialize a boot disk

```
# /etc/vx/bin/vxdisksetup –iB cXtYdZ
```

NOTE: The **–B** option initializes the VERITAS Volume Manager private region to begin at block number 2144. This block is designated as the private region offset for a VERITAS Volume Manager root disk. Without this option, the private region is initialized to start at the default block number 128.

Initialize a data or non-boot disk

```
# /etc/vx/bin/vxdisksetup –i cXtYdZ
```

NOTE:

```
# vxprint –g <dg_name> -d –F “%da_name %privoffset”
```

will tell you which of the vxdisksetup commands listed above was used to initialize the disks of a diskgroup.

```
# vxprint -g rootdg -d -F "%da_name %privoffset"
c3t15d0 2144
c1t15d0 2144
c4t0d2 128
```

you have to reinitialize c4t0d2 before you can use it as a boot disk

Uninitialize a disk: free disk pool → uninitialized disk

```
# /etc/vx/bin/vxdiskunsetup –C cXtYdZ
```

removes private and public regions on a disk and removes the disk access record from the VxVM configuration database.

NOTE: It is not possible to uninitialize a disk that is used in a diskgroup:

```
# /etc/vx/bin/vxdiskunsetup -C c4t0d2
vxdiskunsetup: c4t0d2: Disk device is in use
# vxdg –g <dg_name> rmdisk <dm_name>
# /etc/vx/bin/vxdiskunsetup –C c4t0d2
```

Adding a disk

```
# /usr/sbin/vxdiskadd cXtYdZ
```

Low level command line utility that allows a disk to be added to a diskgroup (ASCII-interface)

Removing a Disk from VxVM

```
# vxdg –g <dg_name> rmdisk <dm_name>
# vxdisk rm <da_name>
```

Create a new diskgroup (the disk should be initialized)

```
#  vxdg init <dg_name> <dm_name>=<da_name>

# vxdg init ccdg ccdg01=c4t0d0
ccdg    = dg_name
ccdg01  = medianame (dm_name)
c4t0d0  = accessname(da_name)

# vxdg list ccdg
Group:     ccdg
dgid:      1042214713.2075.loopback
import-id: 0.2074
flags:
version:   90
local-activation: read-write
detach-policy: global
copies:    nconfig=default nlog=default
config:    seqno=0.1027 permlen=727 free=726 templen=1 loglen=110
config disk c4t0d0 copy 1 len=727 state=clean online
log disk c4t0d0 copy 1 len=110
```

Add a further (initialized) disk to the diskgroup "rootdg"

```
# vxdg –g ccdg adddisk ccdg02=c4t0d1
ccdg02 = medianame (DM)
c4t0d1 = accessname(DA)

# vxdisk list
DEVICE       TYPE      DISK         GROUP        STATUS
c1t15d0      simple    rootdisk02   rootdg       online
c3t15d0      simple    rootdisk01   rootdg       online
c4t0d0       simple    ccdg01       ccdg         online
c4t0d1       simple    ccdg02       ccdg         online
```

Add further disks:

```
# vxdg –g ccdg adddisk ccdg02=c4t0d1 ccdg03=cXtYdZ ..
ccdg0X = medianame (dm_name)
cXtYdZ = accessname(da_name)
```

Rename a diskgroup

```
# vxdg –n <new_dg_name> deport <old_dg_name>
# vxdg import <new_dg_name>
```

Rename a disk in a diskgroup

```
# vxedit –g <dg_name> rename <dm_name_old> <dm_name_new>
```

Remove disks from the diskgroup

```
# vxdg –g <dg_name> rmdisk <dm_name>

# vxdg -g ccdg rmdisk ccdg02
ccdg02 = medianame (dm_name)

# vxdisk list
DEVICE       TYPE      DISK         GROUP        STATUS
c1t15d0      simple    rootdisk02   rootdg       online
c3t15d0      simple    rootdisk01   rootdg       online
c4t0d0       simple    ccdg01       ccdg         online
```

Evacuate the subdisk from one of the disk in the volume

```
# /etc/vx/bin/vxevac –g <dg_name> <dm_name_old> <dm_name_new>

# /etc/vx/bin/vxevac –g ccdg ccdg01 ccdg02
ccdg   = dg_name
ccdg01, ccdg02 = medianame (dm_name)
```

The data will evacuate from that disk to another available space, because it can only removed if there is no data on it. Existing data has to be evacuated. The last disk in the diskgroup can not be removed, it has to be destroyed. The rootdg should never be destroyed.

Destroy the last disk of a diskgroup

```
# vxdg destroy <dg_name>
```

„vxdg destroy“ removes only the objects from the configuration database:

- host ID
- diskname
- dg_name

In case,that the diskgroup was accidently removed, you can only reimport via the CLI (not `vxdiskadm` and VMSA/VEA), by using the group ID.

Deport of a disk group

```
# vxdg deport <dg_name>

# vxdg deport ccdg
```

```
before the deport: # vxdisk list
DEVICE       TYPE      DISK         GROUP        STATUS
c1t15d0      simple    rootdisk02   rootdg       online
c3t15d0      simple    rootdisk01   rootdg       online
c4t0d1       simple    ccdg01       ccdg         online
c4t0d2       simple    ccdg02       ccdg         online


after the deport: # vxdisk list
# vxdisk list
DEVICE       TYPE      DISK         GROUP        STATUS
c1t15d0      simple    rootdisk02   rootdg       online
c3t15d0      simple    rootdisk01   rootdg       online
c4t0d1       simple    -            -            online
c4t0d2       simple    -            -            online
```

Deport to a specific hostname

Disks are locked to a host and not usable after deport

```
# vxdg –h hostname deport <dg_name >
```

Deport to a specific hostname and a specific diskgroupname

```
# vxdg –n <dg_name_new> -h <hostname_name_new> deport <dg_name_old>
```

Display deported diskgroups

```
# vxdisk  -o alldgs list

DEVICE       TYPE      DISK         GROUP        STATUS
c1t15d0      simple    rootdisk02   rootdg       online
c3t15d0      simple    rootdisk01   rootdg       online
c4t0d1       simple    -           (ccdg)        online
c4t0d2       simple    -           (ccdg)        online
```

NOTE:
use **vxdisk -s list** cXtYtZ in order to see which host owns the disk group, in case of deported disk groups (hostid).

```
# vxdisk -s list c4t0d1
Disk:   c4t0d1
type:   simple
flags:  online ready private autoconfig autoimport
diskid: 1042214588.2072.loopback
dgname: ccdg
dgid:   1042214713.2075.loopback
hostid: hprtlc35
info:   privoffset=128
```

Import of disk groups

```
# vxdg import <dg_name>
```

Import of a disk group and rename the disk group name

```
# vxdg –n <dg_name_new> import <dg_name_old>
```

Import of a disk group and rename the disk group name temporarily

```
# vxdg –t –n <dg_name_new_temp> import <dg_name_old>
```

Import of a disk group and clear the import locks

```
# vxdg -tC -n <dg_name_new> import <dg_name_old>
```

Purpose of the locks is to ensure the dual ported disks (disks that can be accessed by two systems simultaneously).

Reimport a destroyed disk group

(because only the dg_name is in that moment destroyed, not the dgid)

```
# vxdg -f import ---.---.<host>
```

```
NOTE: after the import you will be able to start disabled volumes with:
# vxrecover -g <dg_name> -sb
```

Disk Maintenance

List VM-Disks

```
# vxdisk list
```

```
DEVICE       TYPE      DISK         GROUP        STATUS
c1t15d0      simple    rootdisk02   rootdg       online
c3t15d0      simple    rootdisk01   rootdg       online
c4t0d1       simple    ccdg01       ccdg         online
c4t0d2       simple    ccdg02       ccdg         online
```

NOTE: the `vxdisk -s list` gives a summary of listing and more details.

List disk details

```
# vxdisk list <dm_name>
```

```
# vxdisk list rootdisk01
Device:    c3t15d0
devicetag: c3t15d0
type:      simple
hostid:    hprtlc35
disk:      name=rootdisk01 id=1042879607.1036.hprtlc35
timeout:   30
group:     name=rootdg id=1042879605.1025.hprtlc35
info:      privoffset=2144
flags:     online ready private autoimport imported
pubpaths:  block=/dev/vx/dmp/c3t15d0 char=/dev/vx/rdmp/c3t15d0
version:   2.2
iosize:    min=1024 (bytes) max=256 (blocks)
public:    slice=0 offset=3168 len=17778431
private:   slice=0 offset=2144 len=1024
update:    time=1042882193 seqno=0.9
headers:   0 248
configs:   count=1 len=727
logs:      count=1 len=110
Defined regions:
 config   priv 000017-000247[000231]: copy=01 offset=000000 enabled
 config   priv 000249-000744[000496]: copy=01 offset=000231 enabled
 log      priv 000745-000854[000110]: copy=01 offset=000000 enabled
 lockrgn  priv 000855-000919[000065]: part=00 offset=000000
Multipathing information:
numpaths:   1
c3t15d0 state=enabled
```

Status of a disk

online	=	initialized
online invalid	=	disk is uninitialized
LVM	=	disk is in use of LVM

Volume Maintenance

List Volume Information

```
# vxprint –tg dg_name
```

```
# vxprint -tg ignite
DG NAME         NCONFIG      NLOG     MINORS   GROUP-ID
DM NAME         DEVICE       TYPE     PRIVLEN  PUBLEN   STATE
RV NAME         RLINK_CNT    KSTATE   STATE    PRIMARY  DATAVOLS  SRL
RL NAME         RVG          KSTATE   STATE    REM_HOST REM_DG    REM_RLNK
V  NAME         RVG          KSTATE   STATE    LENGTH   READPOL   PREFPLEX UTYPE
PL NAME         VOLUME       KSTATE   STATE    LENGTH   LAYOUT    NCOL/WID MODE
SD NAME         PLEX         DISK     DISKOFFS LENGTH   [COL/]OFF DEVICE   MODE
SV NAME         PLEX         VOLNAME  NVOLLAYR LENGTH   [COL/]OFF AM/NM    MODE

dg ignite       default      default  5711000  1033136039.4944.grcdg456
dm ignite01     XP480_15     simple   1024     2400048  -
dm ignite02     XP480_16     simple   1024     2400048  -
dm ignite03     XP480_17     simple   1024     2400048  -
dm ignite04     XP480_18     simple   1024     2400048  -
sd ignite01-01  ignite-01-01 ignite01 0        400000   0         XP480_15 ENA
sd ignite01-02  ignite-02-01 ignite01 400000   1999472  0         XP480_15 ENA
sd ignite02-01  ignite-02-01 ignite02 0        2400048  1999472   XP480_16 ENA
sd ignite03-01  ignite-02-01 ignite03 0        2400048  4399520   XP480_17 ENA
sd ignite04-01  ignite-02-01 ignite04 0        2400048  6799568   XP480_18 ENA
pl ignite-01-01 ignite-01    ENABLED  ACTIVE   400000   CONCAT    -        RW
pl ignite-02-01 ignite-02    ENABLED  ACTIVE   9199616  CONCAT    -        RW
v  ignite-01    -            ENABLED  ACTIVE   400000   SELECT    -        fsgen
v  ignite-02    -            ENABLED  ACTIVE   9199616  SELECT    -        fsgen
```

If you like to display all informations about all volumes in a diskgroup use

```
# vxprint –va –g ignite  [ oneliner ]
```
or
```
# vxprint –vm –g ignite  [ Each field is output on a separate line ]
```

If you already know which info you need, please use the –F option

Example:
```
# vxprint -v -g ignite -F "%{name: -15} %kstate %state %update_tid"
ignite-01       ENABLED ACTIVE 0.1138
ignite-02       ENABLED ACTIVE 0.1138
```

Volume Read Policies

Read Policies with morroring

- *round robin*
  VxVM reads each Plex in round robin or alternating fashion.
- *prefer*
  VxVM attempts to distribute all reads to the user selected plex.
- *based on layout*
  VxVM evaluates the layout of the plexes in the volume and determines whether the round robin or the prefered read policy will yield the most simultaneous I/O.

Logging Mechanism

Dirty Region Logging (DRL) for mirrored volumes

DRL is a feature available for mirrored VxVM volumes. **DRL** provides recovery for a mirrored volume after a system failure by tracking those regions that have changed as a result of I/O writes to the mirrored volume. It is similar to the HP Logical Volume Manager (LVM) Mirror Write Cache Consistency feature.

write operation → first to that region – marked dirty
→ than to the plex

During an interruption VxVM recovers only those regions of the volume that are marked dirty in the dirty region log. Log subdisks are used to store the dirty region log of a volume that has DRL enabled. Each log subdisk is associated with one plex of a volume.

RAID5 – Logging (Keyword: RAID–5 log plexes)

Only one RAID-5 plex can exist per RAID-5 volume. Any additional plexes become RAID-5 log plexes, which are used to log information about data and parity being written to the volume. When a volume is created using `vxassist` command, a log plex is created for that volume by default. RAID-5 logs can be mirrored and striped.

Create a volume

```
# vxassist –g <dg_name> make <vol_name> <size> <disk_media_name>
```

```
# vxassist -g ccdg make myone 500m ccdg01

# vxprint myone
Disk group: ccdg

TY NAME         ASSOC        KSTATE   LENGTH   PLOFFS   STATE    TUTIL0  PUTIL0
v  myone        fsgen        ENABLED  512000   -        ACTIVE   -       -
pl myone-01     myone        ENABLED  512000   -        ACTIVE   -       -

# vxprint -tg ccdg         (the -t Option shows greater Details)
DG NAME         NCONFIG      NLOG     MINORS   GROUP-ID
DM NAME         DEVICE       TYPE     PRIVLEN  PUBLEN   STATE
RV NAME         RLINK_CNT    KSTATE   STATE    PRIMARY  DATAVOLS  SRL
RL NAME         RVG          KSTATE   STATE    REM_HOST REM_DG    REM_RLNK
V  NAME         RVG          KSTATE   STATE    LENGTH   READPOL   PREFPLEX UTYPE
PL NAME         VOLUME       KSTATE   STATE    LENGTH   LAYOUT    NCOL/WID MODE
SD NAME         PLEX         DISK     DISKOFFS LENGTH   [COL/]OFF DEVICE   MODE
SV NAME         PLEX         VOLNAME  NVOLLAYR LENGTH   [COL/]OFF AM/NM    MODE

dg ccdg         default      default  1104000  1048161290.7640.grcdg456
dm ccdg01       c11t1d0      simple   2000     2399072  -
dm ccdg02       c11t1d1      simple   2000     2399072  -
dm ccdg03       c11t1d2      simple   2000     2399072  -
dm ccdg04       c11t1d3      simple   2000     2399072  -
dm ccdg05       c11t1d4      simple   2000     2399072  -
dm ccdg06       c11t1d5      simple   2000     2399072  -
sd ccdg01-01    myone-01     ccdg01   0        512000   0         c11t1d0  ENA
pl myone-01     myone        ENABLED  ACTIVE   512000   CONCAT    -        RW
v  myone        -            ENABLED  ACTIVE   512000   SELECT    -        fsgen
```

A **default** created volume has always **concatenated** layout.

Create a volume with a striped layout

```
# vxassist -g <dg_name> make <vol_name> <size> layout=stripe\
           ncols=<num_columns> stripeunit=<size> [<disk_media_name> ...]

(columns=<num_columns> for ncols=<num_columns> works two)
```

Example:

```
# vxassist -g ccdg make strvol 300m layout=stripe ncols=6

# vxprint -tg ccdg
dg ccdg         default      default  1104000  1048161290.7640.grcdg456
dm ccdg01       c11t1d0      simple   2000     2399072  -
dm ccdg02       c11t1d1      simple   2000     2399072  -
dm ccdg03       c11t1d2      simple   2000     2399072  -
dm ccdg04       c11t1d3      simple   2000     2399072  -
dm ccdg05       c11t1d4      simple   2000     2399072  -
dm ccdg06       c11t1d5      simple   2000     2399072  -
sd ccdg01-01    strvol-01    ccdg01   0        51200    0/0       c11t1d0  ENA
sd ccdg02-01    strvol-01    ccdg02   0        51200    1/0       c11t1d1  ENA
sd ccdg03-01    strvol-01    ccdg03   0        51200    2/0       c11t1d2  ENA
sd ccdg04-01    strvol-01    ccdg04   0        51200    3/0       c11t1d3  ENA
sd ccdg05-01    strvol-01    ccdg05   0        51200    4/0       c11t1d4  ENA
sd ccdg06-01    strvol-01    ccdg06   0        51200    5/0       c11t1d5  ENA
pl strvol-01    strvol       ENABLED  ACTIVE   307200   STRIPE    6/64     RW
v  strvol       -            ENABLED  ACTIVE   307200   SELECT    strvol-01 fsgen

# vxassist -g ccdg make strvol+log 300m layout=stripe,log ncols=6

# vxprint -tg ccdg | grep strvol+log
sd ccdg01-02    strvol+log-02 ccdg01  0        33       LOG       c11t1d0  ENA
sd ccdg01-01    strvol+log-01 ccdg01  33       51200    0/0       c11t1d0  ENA
sd ccdg02-01    strvol+log-01 ccdg02  0        51200    1/0       c11t1d1  ENA
sd ccdg03-01    strvol+log-01 ccdg03  0        51200    2/0       c11t1d2  ENA
sd ccdg04-01    strvol+log-01 ccdg04  0        51200    3/0       c11t1d3  ENA
sd ccdg05-01    strvol+log-01 ccdg05  0        51200    4/0       c11t1d4  ENA
sd ccdg06-01    strvol+log-01 ccdg06  0        51200    5/0       c11t1d5  ENA
pl strvol+log-01 strvol+log  ENABLED  ACTIVE   307200   STRIPE    6/64     RW
pl strvol+log-02 strvol+log  ENABLED  ACTIVE   LOGONLY  CONCAT    -        RW
v  strvol+log   -            ENABLED  ACTIVE   307200   SELECT    strvol+log-01
fsgen
```

Create a volume with a RAID-5 layout

```
# vxassist -g <dg_name> make <vol_name> <size> layout=raid5\
           ncol=<num_columns> stripeunit=<size> [<disk_media_name> ...]
```

with a log: `ncol = max columns - 1`, (min number of disk = 4)
without a log: `ncol = max columns` , than use `ncol=<num_columns>,nolog`
(min number of disk = 3)
Attention : you have to specify this value if you like to use more than the default number of disks

The stripsize (see `NCOL/WID`) default is 64K (128 sectors) --> the defaults are generally created and defined in `/etc/default/vxassist`.

Example: ccdg with 6 disks

```
# vxassist -g ccdg make raid5 3000m layout=raid5 ncols=6
```

```
vxvm:vxassist: ERROR: Cannot allocate space for 3072000 block volume

# vxassist -g ccdg make raid5 3000m layout=raid5 ncols=5

# vxprint -tg ccdg
DG NAME         NCONFIG      NLOG     MINORS   GROUP-ID
DM NAME         DEVICE       TYPE     PRIVLEN  PUBLEN   STATE
RV NAME         RLINK_CNT    KSTATE   STATE    PRIMARY  DATAVOLS  SRL
RL NAME         RVG          KSTATE   STATE    REM_HOST REM_DG    REM_RLNK
V  NAME         RVG          KSTATE   STATE    LENGTH   READPOL   PREFPLEX UTYPE
PL NAME         VOLUME       KSTATE   STATE    LENGTH   LAYOUT    NCOL/WID MODE
SD NAME         PLEX         DISK     DISKOFFS LENGTH   [COL/]OFF DEVICE   MODE
SV NAME         PLEX         VOLNAME  NVOLLAYR LENGTH   [COL/]OFF AM/NM    MODE

dg ccdg         default      default  1104000  1048161290.7640.grcdg456
dm ccdg01       c11t1d0      simple   2000     2399072  -
dm ccdg02       c11t1d1      simple   2000     2399072  -
dm ccdg03       c11t1d2      simple   2000     2399072  -
dm ccdg04       c11t1d3      simple   2000     2399072  -
dm ccdg05       c11t1d4      simple   2000     2399072  -
dm ccdg06       c11t1d5      simple   2000     2399072  -
sd ccdg01-01    raid5-01     ccdg01   0        768000   0/0       c11t1d0  ENA
sd ccdg02-01    raid5-01     ccdg02   0        768000   1/0       c11t1d1  ENA
sd ccdg03-01    raid5-01     ccdg03   0        768000   2/0       c11t1d2  ENA
sd ccdg04-01    raid5-01     ccdg04   0        768000   3/0       c11t1d3  ENA
sd ccdg05-01    raid5-01     ccdg05   0        768000   4/0       c11t1d4  ENA
sd ccdg06-01    raid5-02     ccdg06   0        2400     0         c11t1d5  ENA
pl raid5-01     raid5        ENABLED  ACTIVE   3072000  RAID      5/16     RW
pl raid5-02     raid5        ENABLED  LOG      2400     CONCAT    -        RW
v  raid5        -            ENABLED  ACTIVE   3072000  RAID      -        raid5
```

Create a volume without a certain disk

```
# vxassist -g dgname make <volumename> <size> layout=stripe
stripeunit=<size> \
ncols=<num_columns> ! [<disk_media_name> ...]
```

Create a volume with mirrored layout

```
# vxassist -g <dg_name> make <vol_name> <size> layout=mirror
```

Example:

```
# vxassist -g ccdg make myone 100m layout=mirror

dg ccdg         default      default  2306000  1042214713.2075.loopback
dm ccdg01       c4t0d1       simple   1024     25163136 -
dm ccdg02       c4t0d2       simple   1024     25163136 -
sd ccdg01-01    myone-01     ccdg01   0        102400   0         c4t0d1   ENA
sd ccdg02-01    myone-02     ccdg02   0        102400   0         c4t0d2   ENA
pl myone-01     myone        ENABLED  ACTIVE   102400   CONCAT    -        RW
pl myone-02     myone        ENABLED  ACTIVE   102400   CONCAT    -        RW
v  myone        -            ENABLED  ACTIVE   102400   SELECT    -        fsgen
```

Mirroring an existing volume

A mirror (plex) can be added to an existing volume with the `vxassist` command, as follows:

```
# vxassist –g <dg_name> mirror <volume_name>
```

For example, to create a mirror of the volume `mytest`, use the following command:

```
# vxassist –g ccdg mirror mytest
```

Another way to mirror an existing volume is by first creating a subdisk and a plex , and then associating it with a volume, using the following commands:

```
# vxmake –g <dg_name> sd <name> disk=<medianame> offset=<dm_offset> len=<size>
# vxmake –g <dg_name>  plex <pl_name> sd=<name>
# vxplex –g <dg_name>  att <v_name> <pl_name>
```

Example :
Create a mirror of volume newvol in diskgroup ccdg

```
# vxprint -g ccdg -v
TY NAME         ASSOC        KSTATE   LENGTH   PLOFFS   STATE    TUTIL0  PUTIL0
v  myone        fsgen        ENABLED  102400   -        ACTIVE   -       -
v  newvol       fsgen        ENABLED  102400   -        ACTIVE   -       -

# vxprint –g ccdg -s -F "%{name:-14} %{pl_name:-10} %{dm_name:-10} %{dm_offset}"
ccdg01-01      myone-01   ccdg01     0
ccdg01-02      newvol-01  ccdg01     102400
ccdg02-01      myone-02   ccdg02     0

# vxmake -g ccdg sd ccdg02-02 disk=ccdg02 offset=102400 len=102400
# vxmake -g ccdg plex newvol-02 sd=ccdg02-02
# vxplex -g ccdg att newvol newvol-02

# vxprint –g ccdg -s -F "%{name:-14} %{pl_name:-10} %{dm_name:-10} %{dm_offset}"
ccdg01-01      myone-01   ccdg01     0
ccdg01-02      newvol-01  ccdg01     102400
ccdg02-01      myone-02   ccdg02     0
ccdg02-02      newvol-02  ccdg02     102400
```

Create a root mirror

```
# /etc/vx/bin/vxrootmir -v –b cXtYdZ
if no full license is installed this command mirrors only the volumes
required to boot successfully from the new mirror.
```

Create a mirror-concat layout

```
# vxassist -g <dg_name> make <vol_name> <size> layout=mirror-concat

  mirror-concat : Specifies that new volumes should be mirrored.  The
                  mirroring is done at the volume level.
```

```
# vxassist -g ccdg make mir_concat 100m layout=mirror-concat

dg ccdg         default      default  1104000  1049968440.7996.grcdg456
dm ccdg01       c11t1d0      simple   1024     2400048  -
dm ccdg02       c11t1d1      simple   2000     2399072  -
dm ccdg03       c11t1d2      simple   2000     2399072  -
dm ccdg04       c11t1d3      simple   2000     2399072  -
dm ccdg05       c11t1d4      simple   2000     2399072  -
dm ccdg06       c11t1d5      simple   2000     2399072  -
sd ccdg01-01    mir_concat-01 ccdg01  0        102400   0         c11t1d0  ENA
sd ccdg02-01    mir_concat-02 ccdg02  0        102400   0         c11t1d1  ENA
pl mir_concat-01 mir_concat  ENABLED  ACTIVE   102400   CONCAT    -        RW
pl mir_concat-02 mir_concat  ENABLED  ACTIVE   102400   CONCAT    -        RW
v  mir_concat   -            ENABLED  ACTIVE   102400   SELECT    -        fsgen
```

Create a concatenated-pro layout (concat-mirror)

```
# vxassist -g <dg_name> make <vol_name> <size> layout=concat-mirror
```

```
concat-mirror : Specifies that new volumes should be concatenated and
                mirrored. The mirroring is handled at each subdisk level.
```

Example:

```
# vxassist -g ccdg make concat_mir 100m layout=concat-mirror

dg ccdg         default      default  1104000  1049968440.7996.grcdg456
dm ccdg01       c11t1d0      simple   1024     2400048  -
dm ccdg02       c11t1d1      simple   2000     2399072  -
dm ccdg03       c11t1d2      simple   2000     2399072  -
dm ccdg04       c11t1d3      simple   2000     2399072  -
dm ccdg05       c11t1d4      simple   2000     2399072  -
dm ccdg06       c11t1d5      simple   2000     2399072  -
sd ccdg01-02    concat_mir-P01 ccdg01 0        102400   0         c11t1d0  ENA
sd ccdg02-02    concat_mir-P02 ccdg02 0        102400   0         c11t1d1  ENA
sd concat_mir-S01 concat_mir-03 concat_mir-L01 0 102400 0        -        ENA
pl concat_mir-03 concat_mir  ENABLED  ACTIVE   102400   CONCAT    -        RW
pl concat_mir-P01 concat_mir-L01 ENABLED ACTIVE 102400  CONCAT    -        RW
pl concat_mir-P02 concat_mir-L01 ENABLED ACTIVE 102400  CONCAT    -        RW
v  concat_mir   -            ENABLED  ACTIVE   102400   SELECT    -        fsgen
v  concat_mir-L01 -          ENABLED  ACTIVE   102400   SELECT    -        fsgen
```

Create striped-mirrored layout

```
# vxassist -g <dg_name> make <vol_name> <size>  layout=stripe-mirror
```

```
# vxassist -g ccdg make str_mir 1000m layout=stripe-mirror

dg ccdg         default      default  1104000  1049968440.7996.grcdg456
dm ccdg01       c11t1d0      simple   1024     2400048  -
dm ccdg02       c11t1d1      simple   2000     2399072  -
dm ccdg03       c11t1d2      simple   2000     2399072  -
dm ccdg04       c11t1d3      simple   2000     2399072  -
dm ccdg05       c11t1d4      simple   2000     2399072  -
dm ccdg06       c11t1d5      simple   2000     2399072  -
sd ccdg01-02    str_mir-P01  ccdg01   0        341376   0         c11t1d0  ENA
sd ccdg02-02    str_mir-P03  ccdg02   0        341376   0         c11t1d1  ENA
sd ccdg03-02    str_mir-P05  ccdg03   0        341376   0         c11t1d2  ENA
sd ccdg04-02    str_mir-P02  ccdg04   0        341376   0         c11t1d3  ENA
sd ccdg05-02    str_mir-P04  ccdg05   0        341376   0         c11t1d4  ENA
sd ccdg06-02    str_mir-P06  ccdg06   0        341376   0         c11t1d5  ENA
sd str_mir-S01  str_mir-03   str_mir-L01 0     341376   0/0       -        ENA
sd str_mir-S02  str_mir-03   str_mir-L02 0     341376   1/0       -        ENA
sd str_mir-S03  str_mir-03   str_mir-L03 0     341376   2/0       -        ENA
pl str_mir-03   str_mir      ENABLED  ACTIVE   1024128  STRIPE    3/64     RW
pl str_mir-P01  str_mir-L01  ENABLED  ACTIVE   341376   CONCAT    -        RW
pl str_mir-P02  str_mir-L01  ENABLED  ACTIVE   341376   CONCAT    -        RW
pl str_mir-P03  str_mir-L02  ENABLED  ACTIVE   341376   CONCAT    -        RW
pl str_mir-P04  str_mir-L02  ENABLED  ACTIVE   341376   CONCAT    -        RW
pl str_mir-P05  str_mir-L03  ENABLED  ACTIVE   341376   CONCAT    -        RW
pl str_mir-P06  str_mir-L03  ENABLED  ACTIVE   341376   CONCAT    -        RW
v  str_mir      -            ENABLED  ACTIVE   1024000  SELECT    str_mir-03 fsgen
v  str_mir-L01  -            ENABLED  ACTIVE   341376   SELECT    -        fsgen
v  str_mir-L02  -            ENABLED  ACTIVE   341376   SELECT    -        fsgen
v  str_mir-L03  -            ENABLED  ACTIVE   341376   SELECT    -        fsgen
```

Remove a Volume

Step 1. Remove all references to the volume.

Step 2. If the volume is mounted as a file system, unmount it with the command:
`# umount /dev/vx/dsk/ volume_name`

Step 3. If the volume is listed in `/etc/fstab`, remove its entry.

Step 4. Make sure that the volume is stopped with the command:
`# vxvol stop volume_name`
The `vxvol stop` command stops all VM activity to the volume. After following these steps, remove the volume with one of the following commands:

```
# vxedit -g <dg_name> -rf rm <vol_name>
```

The **`-r`** option indicates recursive removal, which means the removal of all plexes associated with the volume and all subdisks associated with those plexes. The `-r` option of the `vxedit` command removes multiple objects.

The **`-f`** option forces removal, and is necessary if the volume is enabled.

You can also remove an entire volume with the `vxassist` command. Use the keywords `remove` and `volume` and provide the volume name on the command line as shown in the following example:

```
# vxassist -g <dg_name> remove volume <vol_name>
```

Devices for the generated Volumes

```
# /dev/vx/rdsk/dg_name/vol_name
```

Example:

```
VxVM 3.1 : # ll /dev/vx/rdsk/peter/volume1
crw-------   1 root       root        99 0x2834f8 Jan 11 22:34
/dev/vx/rdsk/peter/volume1

VxVM 3.2 : # ll /dev/vx/rdsk/ccdg/myone
crw-------   1 root       root        63 0x10d880 Apr 13 14:56
/dev/vx/rdsk/ccdg/myone

VxVM 3.5 : # ll /dev/vx/rdsk/rootdg/varvol
crw-------   1 root       root        33 0x00000a Jan 15 15:52
/dev/vx/rdsk/rootdg/varvol


# newfs -F vxfs /dev/vx/rdsk/ccdg/myone
    version 4 layout
    102400 sectors, 102400 blocks of size 1024, log size 1024 blocks
    unlimited inodes, largefiles not supported
    102400 data blocks, 101280 free data blocks
    4 allocation units of 32768 blocks, 32768 data blocks
    last allocation unit has 4096 data blocks
```

Change permissions of the device

```
# vxedit -g <dg_name> set mode=rwx <vol_name>
```

Example:

```
# vxedit -g ccdg set mode=666 myone
```

Check the permissions of the volume

```
# vxprint –g <dg_name> -l <vol_name> | grep perms
```

```
# vxedit –g ccdg set mode=666 myone
# vxprint –g ccdg -l myone | grep perms
perms:    user=root group=root mode=0666
```

Check the read policy

```
# vxprint –g <dg_name> -l <vol_name>
```

```
# vxprint –g ccdg -l myone | grep policies
policies: read=SELECT (round-robin) exceptions=GEN_DET_SPARSE
```

Change the read policy (round robin, prefer, based on layout)

```
vxvol rdpol <policy> vol_name
```

Example:

```
# vxprint -g ccdg -l myone | grep policies
policies: read=SELECT (prefer myone-03) exceptions=GEN_DET_SPARSE

# vxvol rdpol round myone

# vxprint -g ccdg -l myone | grep policies
policies: read=ROUND exceptions=GEN_DET_SPARSE

policy: round, prefer, select
```

Add a comment to the plex

```
vxedit set comment="string" <plex_name>
```

```
# vxedit set comment="cogito ergo sum" myone-03
# vxprint -l myone-03
Disk group: ccdg

Plex:     myone-03
info:     len=1024128
type:     layout=STRIPE columns=3 width=64
state:    state=ACTIVE kernel=ENABLED io=read-write
assoc:    vol=myone sd=myone-S01,myone-S02,myone-S03
flags:    complete
comment:  cogito ergo sum
```

Internals

How to read the header information of a disk

xd – command

Example (VxVM 3.1 / 3.2 / 'non-bootable 3.5 disk'):

```
# xd -tc -j128k -N8 /dev/rdsk/c11t1d0
0000000    P   R   I   V   H   E   A   D

# xd -tc -j131116 -N24 /dev/rdsk/c11t1d0
0000000    1   0   4   9   9   6   5   1   1   3   .   7   9   8   7   .
0000010    g   r   c   d   g   4   5   6

# xd -tc -j131244 -N24 /dev/rdsk/c11t1d0
0000000    1   0   4   9   9   6   8   4   4   0   .   7   9   9   6   .
0000010    g   r   c   d   g   4   5   6
```

compared with the

```
# vxdisk list c11t1d0 | grep -e group: -e disk:
disk:      name=ccdg01 id=1049965113.7987.grcdg456
group:     name=ccdg id=1049968440.7996.grcdg456
```

Example (VxVM 3.5 / bootable rootdg):

```
# xd -tc -j2144k -N8 /dev/rdsk/c0t6d0
0000000    P   R   I   V   H   E   A   D

# xd -tc -j2195500 -N24 /dev/rdsk/c0t6d0
0000000    1   0   4   2   6   4   0   0   5   0   .   1   0   2   8   .
0000010    g   r   c   d   g   1   1   9

# xd -tc -j2195628 -N24 /dev/rdsk/c0t6d0
0000000    1   0   4   2   6   4   0   0   4   9   .   1   0   2   5   .
0000010    g   r   c   d   g   1   1   9

# vxdisk list c0t6d0 | grep -e group: -e disk:
disk:      name=rootdisk01 id=1042640050.1028.grcdg119
group:     name=rootdg id=1042640049.1025.grcdg119
```

vxprivutil

```
# /etc/vx/diag.d/vxprivutil scan /dev/dsk/c0t6d0
diskid:  1042640050.1028.grcdg119
group:   name=rootdg id=1042640049.1025.grcdg119
flags:   private autoimport
hostid:  loopback
version: 2.2
iosize:  1024
public:  slice=0 offset=3168 len=4189088
private: slice=0 offset=2144 len=1024
update:  time: 1050061599  seqno: 0.138
headers: 0 248
configs: count=1 len=727
logs:    count=1 len=110

# /etc/vx/diag.d/vxprivutil dumpconfig /dev/dsk/c0t6d0 | more
#Config copy 01
```

```
#Header nblocks=5816 blksize=128 hdrsize=512
#flags=0x100 (CLEAN)
#version: 4/9
#dgname: rootdg  dgid: 1042640049.1025.grcdg119
#config: tid=0.1236 nrvg=0 nrlink=0 nvol=9 nplex=9 nsd=9 ndm=1 nda=0
#pending: tid=0.1236 nrvg=0 nrlink=0 nvol=9 nplex=9 nsd=9 ndm=1 nda=0
#
#Block    6: flag=0    ref=2    offset=0    frag_size=67
#Block    7: flag=0    ref=15   offset=0    frag_size=67
#Block   10: flag=0    ref=3    offset=0    frag_size=83
#Block   11: flag=0    ref=10   offset=0    frag_size=78
#Block   14: flag=0    ref=4    offset=0    frag_size=104
#Block   15: flag=0    ref=46   offset=0    frag_size=90
#Block   16: flag=0    ref=4    offset=104  frag_size=2
#Block   17: flag=0    ref=47   offset=0    frag_size=70
#Block   19: flag=0    ref=48   offset=0    frag_size=90
#Block   21: flag=0    ref=20   offset=0    frag_size=58
#Block   23: flag=0    ref=49   offset=0    frag_size=69
#Block   25: flag=0    ref=5    offset=0    frag_size=81
#Block   27: flag=0    ref=6    offset=0    frag_size=103
#Block   29: flag=0    ref=26   offset=0    frag_size=77
#Block   30: flag=0    ref=7    offset=0    frag_size=81
#Block   32: flag=0    ref=16   offset=0    frag_size=102
#Block   33: flag=0    ref=50   offset=0    frag_size=90
#Block   35: flag=0    ref=51   offset=0    frag_size=69
#Block   37: flag=0    ref=52   offset=0    frag_size=90
#Block   39: flag=0    ref=53   offset=0    frag_size=69
#Block   41: flag=0    ref=54   offset=0    frag_size=93
#Block   43: flag=0    ref=55   offset=0    frag_size=69
#Block   45: flag=0    ref=44   offset=0    frag_size=77
#Block   46: flag=0    ref=66   offset=0    frag_size=96
#Block   48: flag=0    ref=67   offset=0    frag_size=73
#Block   50: flag=0    ref=68   offset=0    frag_size=59
#Block   63: flag=0    ref=59   offset=0    frag_size=58
#Block   75: flag=0    ref=71   offset=0    frag_size=59
#Block   87: flag=0    ref=83   offset=0    frag_size=59
#Block   99: flag=0    ref=95   offset=0    frag_size=59
#Block  111: flag=0    ref=109  offset=0    frag_size=59
#
#Record    2: type=0xe015 flags=0    gen_flags=0x4  size=67
#Blocks: 6
dg   rootdg
  comment="
  putil0="
  putil1="
  putil2="
  dgid=1042640049.1025.grcdg119
  rid=0.1025
  update_tid=0.1027
  nconfig=default
  nlog=default
  base_minor=0
  version=90
#Record    3: type=0x4012 flags=0    gen_flags=0x4  size=83
#Blocks: 10
plex standvol-01
  comment=""stand plex"
  putil0="
  putil1="
  putil2="
  layout=CONCAT
  ...
```

Recovery

Debug level for vxconfigd (3.2 and 3.5)

The following command logs all debug and error messages to the specified log file :

```
# vxconfigd –k –m enable -x<debuglevel> –x log >/<output_file> 2>&1
```

Example :

```
# vxconfigd –k –m enable –x9 –x log >/tmp/vxconfigd.log 2>&1
```

This command will first kill (-k) an already running vxconfigd process before any other startup processing. This is useful for recovering from a hung vxconfigd process. Killing the old vxconfigd and starting a new one usually does not cause problems for volume devices that are being used by applications, or that contain mounted file systems.

Fixing a missing /etc/vx/volboot

The volboot file contains an initial list of disks that are used to locate the root disk group. It also contains a host ID that is stored on disks in imported disk groups to define ownership of disks as a sanity check for disks that might be accessible from more than one host.

This host ID is used to ensure that two or more hosts that can access disks on a shared SCSI bus will not interfere with each other in their use of those disks.

A method for performing limited recovery on a VxVM boot disk is to use the VxVM Maintenance Mode Boot (MMB). MMB mode is initiated by booting the system and gaining control at the ISL prompt :

```
Boot terminated.


------- Main Menu -----------------------------------------------------------

        Command                          Description
        -------                          -----------
        BOot [PRI|ALT|<path>]            Boot from specified path
        PAth [PRI|ALT|CON|KEY] [<path>] Display or modify a path
        SEArch [DIsplay|IPL] [<path>]    Search for boot devices

        COnfiguration [<command>]        Access Configuration menu/commands
        INformation [<command>]          Access Information menu/commands
        SERvice [<command>]              Access Service menu/commands

        DIsplay                          Redisplay the current menu
        HElp [<menu>|<command>]          Display help for menu or command
        RESET                            Restart the system
-------
Main Menu: Enter command >bo pri ipl
Interact with IPL (Y, N, Q)?> y

Booting...
Boot IO Dependent Code (IODC) revision 1
```

```
HARD Booted.

ISL Revision A.00.43  Apr 12, 2000

ISL> hpux -vm
```

```
VxVM Maintenance Mode boot

Boot
: disk(8/0/19/0.6.0.0.0.0.0;0)/stand/vmunix
8008844 + 786432 + 614752 start 0x13ff68
NOTICE: autofs_link(): File system was registered at index 3.
NOTICE: cachefs_link(): File system was registered at index 5.
NOTICE: nfs3_link(): File system was registered at index 6.

    System Console is on the Built-In Serial Interface
Entering cifs_init...
Initialization finished successfully... slot is 9
    Swap device table:  (start & size given in 512-byte blocks)
        entry 0 - auto-configured on root device; ignored - no room
WARNING: no swap device configured, so dump cannot be defaulted to primary swap.
WARNING: No dump devices are configured.  Dump is disabled.
Starting the STREAMS daemons-phase 1
Starting vxconfigd in boot mode (pre_init_rc).
INFO: VxVM Maintenance Mode Boot - vxconfigd aborted
Checking root file system.
file system is clean - log replay is not required
Root check done.
Create STCP device files
        $Revision: vmunix:    vw: -proj    selectors: CUPI80_BL2000_1108 -c 'Vw for
CUPI80_BL2000_
1108 build' -- cupi80_bl2000_1108 'CUPI80_BL2000_1108'  Wed Nov  8 19:05:38 PST
2000 $
Memory Information:
    physical page size = 4096 bytes, logical page size = 4096 bytes
    Physical: 131072 Kbytes, lockable: 82856 Kbytes, available: 96976 Kbytes

/sbin/ioinitrc:
Can't open /dev/vx/dsk/rootdg/standvol, errno = 6
/dev/vx/dsk/rootdg/standvol: CAN'T CHECK FILE SYSTEM.
/dev/vx/dsk/rootdg/standvol: UNEXPECTED INCONSISTENCY; RUN fsck MANUALLY.
/dev/vx/dsk/rootdg/standvol: No such device or address
Unable to mount /stand - please check entries in /etc/fstab
Skipping KRS database initialization - /stand can't be mounted

INFO: VxVM Maintenance Mode Boot - vxconfigd aborted

INIT: Overriding default level with level 's'

INIT: SINGLE USER MODE

INIT: Running /sbin/sh

# mount -v
/dev/vx/dsk/rootdg/rootvol on / type vxfs log on Mon Nov 11 10:56:18 2002

# vxconfigd -m disable

# vxdctl init loopback

# vxdctl add disk c0t6d0 privoffset=2144

# vxconfigd -kr reset -m boot
```

```
# vxdisk list
DEVICE       TYPE      DISK         GROUP        STATUS
c0t6d0       simple    rootdisk01   rootdg       online

# vxvol startall

# mountall

# mount -v
/dev/vx/dsk/rootdg/rootvol on / type vxfs log on Mon Nov 11 10:56:18 2002
/dev/vx/dsk/rootdg/varvol on /var type vxfs delaylog,nodatainlog on Mon Nov 11
11:11:28 2002
/dev/vx/dsk/rootdg/usrvol on /usr type vxfs delaylog,nodatainlog on Mon Nov 11
11:11:29 2002
/dev/vx/dsk/rootdg/tmpvol on /tmp type vxfs delaylog,nodatainlog on Mon Nov 11
11:11:29 2002
/dev/vx/dsk/rootdg/standvol on /stand type hfs defaults on Mon Nov 11 11:11:29 2002
/dev/vx/dsk/rootdg/optvol on /opt type vxfs delaylog,nodatainlog on Mon Nov 11
11:11:29 2002
/dev/vx/dsk/rootdg/homevol on /home type vxfs delaylog,nodatainlog on Mon Nov 11
11:11:30 2002

# ll /etc/vx/volboot
-rw-r--r--   1 root       root          1024 Nov 11 11:08 volboot

# reboot
```

Missing /stand/ioconfig

Unfortunately the solution described in /usr/share/doc/vxvm/vxvm_tshoot.pdf (page 24p.) does not work. The following solution requires an already configured ignite server, but booting from the recovery-CD should be possible too.

```
Boot terminated.


------- Main Menu -------------------------------------------------------------

        Command                          Description
        -------                          -----------
        BOot [PRI|ALT|<path>]            Boot from specified path
        PAth [PRI|ALT|CON|KEY] [<path>] Display or modify a path
        SEArch [DIsplay|IPL] [<path>]    Search for boot devices

        COnfiguration [<command>]        Access Configuration menu/commands
        INformation [<command>]          Access Information menu/commands
        SERvice [<command>]              Access Service menu/commands

        DIsplay                          Redisplay the current menu
        HElp [<menu>|<command>]          Display help for menu or command
        RESET                            Restart the system
-------
Main Menu: Enter command > bo lan.<ignite-server-ip> INSTALL
Interact with IPL (Y, N, Q)?> N
```

```
                        Welcome to Ignite-UX!

  Use the <tab> key to navigate between fields, and the arrow keys
  within fields.  Use the <return/enter> key to select an item.
  Use the <return> or <space-bar> to pop-up a choices list.  If the
  menus are not clear, select the "Help" item for more information.

  Hardware Summary:          System Model: 9000/778/B132L+
  +---------------------+----------------+-------------------+ [ Scan Again  ]
  | Disks: 1  (  4.0GB) |  Floppies: 0   | LAN cards:   1    |
  | CD/DVDs:        0   |  Tapes:    0   | Memory:     128Mb |
  | Graphics Ports: 1   |  IO Buses: 3   | CPUs:        1    | [ H/W Details ]
  +---------------------+----------------+-------------------+

                       [      Install HP-UX       ]

                       [   Run a Recovery Shell   ]

                       [    Advanced Options      ]


          [  Reboot  ]                              [  Help  ]
```

```
Networking must be enabled in order to load a shell.

(Press any key to continue.)

* Searching the network for a DHCP server to supply default networking
  information....

  This could take up to 30 seconds if a DHCP server cannot be found. If
  you wish to cancel the DHCP server search, you may press CTRL-C now.
```

```
                             NETWORK CONFIGURATION

                                  This system's hostname:

  Internet protocol address (eg. 15.2.56.1) of this host:

       Default gateway routing internet protocol address:

       The subnet mask (eg. 255.255.248.0 or 0xfffff800): 255.255.248.0

                IP address of the Ignite-UX server system: 15.140.11.155

          Is this networking information only temporary?  [ Yes ]



   [   OK   ]                  [ Cancel ]                          [  Help  ]

please fill in and press okay

       * Bringing up Network (lan0)
```

```
       * Loading insf to create disk device files...
       * Creating disk device files...
       * Loading the recovery commands...

(c) Copyright 1999, 2000, 2001, 2002 Hewlett-Packard Co.
(c) Copyright 1979 The Regents of the University of Colorado, a body corporate
(c) Copyright 1979, 1980, 1983 The Regents of the University of California
(c) Copyright 1980, 1984 AT&T Technologies.  All Rights Reserved.



                  HP-UX NETWORK SYSTEM RECOVERY

                  WARNING:  YOU ARE SUPERUSER !!


Checking for required components on the Ignite Server........

Loading some basic commands...



                HP-UX NETWORK SYSTEM RECOVERY
                           MAIN MENU


        s.  Search for a file
        b.  Reboot
        l.  Load a file
        r.  Recover an unbootable HP-UX system
        x.  Exit to shell


This menu is for listing and loading the tools contained on the core media.
Once a tool is loaded, it may be run from the shell. Some tools require other
files to be present in order to successfully execute.

Select one of the above:
```

```
Loading commands for recovery ....



        DEVICE FILE VERIFICATION MENU

    This menu is used to specify the path of the root file system.
    When the information is correct, select 'a'.

        INFORMATION to verify:
          Device file used for '/'(ROOT) is c0t6d0
          The hardware path to disk is 8/0/19/0.6.0

        Select one of the following:
          a.  The above information is correct.
          b.  WRONG!! The device file used for '/'(ROOT) is incorrect.

          m.  Return to the 'HP-UX Recovery MENU.'
          x.  Exit to the shell.

     Selection: a
```

```
The disk is VxVM
Setting boot and root device file for c0t6d0...


                     HP-UX Recovery MENU

        Select one of the following:
        a.  Mount the root disk and exit to a shell only.
        b.  Recover the bootlif/os partitions.
        c.  Replace the kernel on the root file system.
        d.  Both Options: b and c
        v.  Read information about VxVM Recovery

        m.  Return to 'HP-UX Recovery Media Main Menu'.
        x.  Exit to the shell.

    Use this menu to select the level of recovery desired.

    Selection: a
```

```
Entering file system checking ...
Doing Repair/Restore of VxVM volumes

Loading vxprivutil command

Loading vxconfigd command for starting volume manager
vxvm:vxconfigd: WARNING: File /etc/vx/array.info doesn't exist.
Generating file /etc/vx/array.info.

vxvm:vxconfigd: WARNING: File /etc/vx/array.info doesn't exist.
Generating file /etc/vx/array.info.

vxvm:vxconfigd: WARNING: File /etc/vx/array.info doesn't exist.
Generating file /etc/vx/array.info.

Loading fsck command for recovery ....
file system is clean - log replay is not required
standvol is HFS so remove the /sbin/fs/vxfs/fsck command
** /dev/vx/rdsk/rootdg/standvol
** Last Mounted on /stand
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
65 files, 0 icont, 5172 used, 7114 free (130 frags, 873 blocks)
Loading mount Command for recovery ....
Mounting root filesystem on /ROOT


================================================================
The root disk has been mounted under /ROOT, if you want to chroot
to this directory type the commands:
     loadfile chroot
     chroot /ROOT /sbin/sh


please use the commands as proposed
```

```
# loadfile chroot

# chroot /ROOT /sbin/sh

# mount -v
/dev/vx/dsk/rootdg/rootvol on / type vxfs log on Wed Nov  6 12:43:05 2002
/dev/vx/dsk/rootdg/standvol on /stand type hfs defaults on Wed Nov  6 12:43:07 2002
/dev/vx/dsk/rootdg/varvol on /var type vxfs delaylog,nodatainlog on Wed Nov  6
12:43:20 2002
/dev/vx/dsk/rootdg/usrvol on /usr type vxfs delaylog,nodatainlog on Wed Nov  6
12:43:20 2002
/dev/vx/dsk/rootdg/tmpvol on /tmp type vxfs delaylog,nodatainlog on Wed Nov  6
12:43:21 2002
/dev/vx/dsk/rootdg/optvol on /opt type vxfs delaylog,nodatainlog on Wed Nov  6
12:43:21 2002
/dev/vx/dsk/rootdg/homevol on /home type vxfs delaylog,nodatainlog on Wed Nov  6
12:43:21 2002
-hosts on /net type autofs ignore,indirect,nosuid,soft,rsize=32768,wsize=32768 on
Wed Nov  6 12:44
:15 2002

# cd stand
# ls
boot.sys          dlkm.vmunix.prev  krs_tmp           system.d
bootconf          kernrel           lost+found        system.prev
build             krs               rootconf          vmunix
dlkm              krs_lkg           system            vmunix.prev

# cat /etc/ioconfig >/stand/ioconfig

# reboot
```

Disk failure / Disk replacement

Case 1: Partial Disk Failure

If hot-relocation is enabled when a plex or disk is detached by a failure, mail indicating the failed objects is sent to `root`. If a partial disk failure occurs, the mail identifies the failed plexes. For example, if a disk containing mirrored volumes fails, mail information is sent as shown in the following display:

```
To: root
Subject: Volume Manager failures on host teal
Failures have been detected by the VERITAS Volume
Manager:
failed plexes:
home-02
src-02
```

To determine which disk is causing the failures in the above example message, enter the following command:

```
# vxstat -s -ff home-02 src-02
```

A typical output display is as follows:

```
FAILED
TYP NAME READS WRITES
sd disk01-04 0 0
sd disk01-06 0 0
```

```
sd disk02-03 1 0
sd disk02-04 1 0
```

This display indicates that the failures are on `disk02` (and that subdisks `disk02-03` and `disk02-04` are affected). Hot-relocation automatically relocates the affected subdisks and initiates any necessary recovery procedures. However, if relocation is not possible or the hot-relocation feature is disabled, you have to investigate the problem and attempt to recover the plexes. These errors can be caused by cabling failures, so check the cables connecting your disks to your system. If there are obvious problems, correct them and recover the plexes with the following command:

```
# vxrecover -b home src
```

This command starts recovery of the failed plexes in the background (the command returns before the operation is done). If an error message appears later, or if the plexes become detached again and there are no obvious cabling failures, replace the disk (see next section).

Case 2: Complete Disk Failure – Replacing a failed Disk

If a disk fails completely and hot-relocation is enabled, the mail message lists the disk that failed and all plexes that use the disk. For example, mail information is sent as shown in the following display:

```
To: root
Subject: Volume Manager failures on host teal
Failures have been detected by the VERITAS Volume
Manager:
failed disks:
disk02
failed plexes:
home-02
src-02
mkting-01

failing disks:
disk02
```

This message shows that `disk02` was detached by a failure. When a disk is detached, I/O cannot get to that disk. The plexes `home-02`, `src-02`, and `mkting-01` are also detached because of the disk failure. Again, the problem can be a cabling error. If the problem is not a cabling error, replace the disk.

Replacing a failed disk

(see also Recovery 3.5 : Recovering from a failed VxVM Boot Mirror Disk)

1. **If the disk is still alive, but has to be replaced**, use `vxdiskadm : Select menu item 3 (Remove a disk for replacement)`. first.

2. **Disabling a disk**
   You can take a disk offline. If the disk is corrupted, you need to take it offline and remove it. You may be moving the physical disk device to another location to be connected to another system. To take a disk offline, first remove it from its disk group, and then use the following procedure:

```
vxdiskadm:  Select menu item 10 (Disable (offline) a disk device)
```

3. **Replacing a disk**
   If a disk was replaced due to a disk failure and you wish to move hot-relocated subdisks back to this replaced disk , see chapter 9, Administrator Guide

   ```
   vxdiskadm: Select menu item 4 (Replace a failed or removed disk)
   ```

   When the drive is failed and was replaced by a new drive , the process will create a new public and private region (if needed) and populate the private region with the disk media name of the failed disk. Then it will run `vxreattach -r` (tries to recover stale plexes of any volumes on the failed disk) and `vxrecover -s` (starts disabled volumes that are selected by the operation).

Reattaching Disks

In case a disk has a **full failure** and **hot-relocation is not possible**, or you have some missing disk drivers, you can use after the `vxreattach` command to reattach the disks without plexes being flagged as stale.
`vxreattach` reattaches the failed disk media record to the disk with the same device name.
`vxreattach -c` checks whether a reattach is possible.

Recovering the Volume Manager Configuration

Once the Volume Manager package has been loaded, recover the Volume Manager configuration using the following procedure:

1) Touch `/etc/vx/reconfig.d/state.d/install-db`.
   If the system is rebooted and the `install-db` file exists, VxVM is not activated at system startup.
2) Shut down the system.
3) Reattach the disks that were removed from the system.
4) Reboot the system.
5) When the system comes up, bring the system to single-user mode:
   ```
   # exec init S
   ```
6) When prompted enter the password and press Return to continue.
   NOTE:
   ```
   vxdisklist: "vxvm. IPC failure. Configuration daemon is non accessible"
   ```
7) Remove files involved with installation that were created when you loaded Volume Manager but are no longer needed:
   ```
   # rm -rf /etc/vx/reconfig.d/state.d/install-db
   ```
8) Start some Volume Manager I/O daemons:
   ```
   # vxiod set 10
   ```
9) Start the Volume Manager configuration daemon, `vxconfigd`, in disabled mode:
   ```
   # vxconfigd -m disable
   ```
10) Initialize the `vxconfigd` daemon:
    ```
    # vxdctl init
    ```

11) Initialize the DMP subsystem:
```
# vxdctl initdmp
```
This creates user level nodes for all the DMP devices that have been detected by the kernel. This will remove all the existing DMP nodes in `/dev/vx[r]dmp` directory, and create fresh nodes for the DMP devices that have been detected.

12) Enable `vxconfigd`:
```
# vxdctl enable
```

See also http://www.docs.hp.com/hpux/onlinedocs/5187-1374/5187-1374.html

Recovering from a failed VxVM Boot Mirror Disk (VxVM 3.5)

If a failed primary boot disk is under VxVM control and is mirrored, you follow one of the following replacement scenarios.

Interactive way

1. If the disk is still alive, but has to be replaced, use vxdiskadm command option 3 (Remove a disk for replacement) first.
2. Replace the failed boot disk. Depending on the system hardware, this may require you to shut down and power off the system
3. Boot the system from mirror of the root disk, and use the following command to initialize the replacement disk :
   # /etc/vx/bin/vxdisksetup –iB cXtYdZ privlen=1024
4. Run the vxdiskadm command, and use option 4 (Replace a failed or removed disk) to define the newly initialized disk as the replacement for the rootdisk. The vxdiskadm starts a vxrecover which needs some time. Use the following command to get the status :
```
# vxtask list
TASKID  PTID TYPE/STATE    PCT   PROGRESS
   165          PARENT/R 75.00% 8/6(1) VXRECOVER
   165   165    ATCOPY/R 20.32% 0/1093632/222208 PLXATT usrvol usrvol-02
```
5. After vxrecover has finished use the vxbootsetup command to make the disk bootable
```
# /etc/vx/bin/vxbootsetup rootdisk02
```

6. Verify the contents of the updated LABEL file:
```
# vxvmboot –v /dev/rdsk/cXtYdZ
LIF Label File @ (1k) block # 1178 on VxVM Disk /dev/rdsk/c1t15d0:
Label Entry: 0, Boot Volume start:     3168; length: 300 MB
Label Entry: 1, Root Volume start:  2407520; length: 200 MB
Label Entry: 2, Swap Volume start:   310368; length: 2048 MB
Label Entry: 3, Dump Volume start:   310368; length: 2048 MB
```

Step-by-step

1. Replace the failed boot disk. Depending on the system hardware, this may require you to shut down and power off the system
2. Boot the system from mirror of the root disk, and use the following command to initialize the replacement disk:

```
# /etc/vx/bin/vxdisksetup –iB cXtYdZ privlen=1024
# vxdg –g <dg_name> -k rmdisk rootdisk02
# vxdg –g <dg_name> -k adddisk rootdisk02=cXtYdZ
```

```
# vxrecover  -g <dg_name> -v
# mkboot -l
# vxvmboot  -b -o 3168 -l 307200 /dev/rdsk/cXtYdZ
# vxvmboot  -r -o 2407520 -l 204800 /dev/rdsk/cXtYdZ
# vxvmboot  -s -o 310368 -l 2097152 /dev/rdsk/cXtYdZ
# vxvmboot  -d -o 310368 -l 2097152 /dev/rdsk/cXtYdZ
# vxvmboot  –v /dev/rdsk/cXtYdZ
```

How to recover rootdg that had only 1 disk after a disk failure (VxVM 3.2)

(see KMine Doc VXVMKBRC00006345)

The following steps can be used to recreate a rootdg that had only one disk after that disk failed. This document assumes that the original disk in the rootdg and the replacment disk are both `/dev/rdsk/c3t1d0`.

1) Replace the failed disk and boot the system

2) Set the vxconfigd to run in disabled mode
```
# vxconfigd -m disable
```
If the following error is returned:
```
vxvm:vxconfigd: ERROR: cannot open /dev/vx/config: Device is already open
```
check the state of vxconfigd as follows:
```
# vxdctl mode
```
The output should look like this:
```
mode: disabled
```

3) Start 10 `vxiod` processes:
```
# vxiod set 10
```

4) Run vxconfigd in disabled mode, and create the necessary `/dev/vx/dmp` device files:
```
# vxdctl init
```

5) Create rootdg with no disks:
```
# vxdg init rootdg
```

6) Initialize the original disk for use in the rootdg:
```
# vxdisk init c3t1d0
```

7) Add the disk to rootdg:
```
vxdg adddisk disk01=c3t1d0
```

8) Reset vxconfigd to run in normal mode, and bring the other disk groups back online:
```
# vxdctl enable
```

9) Recover the rootdg volumes from the dgcfgbackup file. Default is `/etc/vxvmconf/dg_name.conf`; create by `/usr/sbin/dgcfgbackup [-f dg_conf_path] dg_name`:
```
# dgcfgrestore -n rootdg c3t1d0
```

10) At this point the volumes are recreated with their original sizes, but are not yet enabled. Run the following command for each volume in the rootdg:
```
# vxvol –g rootdg start vol01
```

11) Start all of the other volumes:
```
# vxvol startall
```

12) Now any volumes in the rootdg that contained filesystems will have to have those filesystems recreated:
```
# newfs /dev/vx/rdsk/rootdg/vol01
```

13) Then the all of the volumes will have to be mounted:
```
# mount -a
```

And finally any data in the rootdg volumes/file sytems will have to be recovered from archives. Now the rootdg is recovered, and VxVM will start automatically at each reboot.

Dynamic Multipathing (DMP)

You may need an **additional license** to use this feature. In a clustered environment where Active/Passive (A/P) type disk arrays are shared by multiple hosts, all hosts in the cluster should access the disk via the same physical path. If a disk from an Active/Passive type shared disk array is accessed via multiple paths simultaneously, it could lead to severe degradation of I/O performance. This requires path failover on a host to be a cluster coordinated activity for an Active/Passive (A/P) type disk array.
For **Active/Active (A/A) type disk arrays**, any disk can be simultaneously accessed through all available physical paths to it. Therefore, in a clustered environment all hosts do not need to access a disk, via the same physical path.

Questions & Answers

Q: What is the difference between VxVM Dynamic Multipathing (**DMP**) and the LVM Physical Volume Links (**PV links**)?

A: Both VxVM DMP and LVM PV links allow you to set alternate links to a physical volume (disk). Path switching takes place automatically on disk failure or manually using specific commands. PV links have to be created by the user whereas DMP is always active (for Active/Active disk arrays). DMP provides load balancing (for Active/Active disk arrays) whereas PV links does not.

Q: How many paths can i have to one disk device with DMP?

A: There is no limit.

Q: When using DMP with a XP256 disk array, what is the host mode that I should use for the ports connected to the system?

A: The XP256 host mode should be set to mode 08 if you are using DMP. Mode 08 allows the system to see all logical units (LUNs). Mode 00 only allows the system to recognize eight LUNs.
NOTES:
If you have Hitachi Data Systems (HDS) 7700E and 9900 arrays, the system will only see eight LUNs, even if you use host mode 08.
Mode 09 is intended for use on Sun Solaris systems.

DMP and EMC Power Path Exclusive Use Support

DMP co-exists with Power Path by allowing Power Path to manage the I/O to the disk array. At the present time it is not possible to have DMP manage some LUNs and Power Path manage others on the same disk array. Therefore, either DMP or Power Path has exclusive use of the disk array. There are no known problems with having DMP and Power Path installed and running on the same host. DMP simply passes the I/O to Power Path.
There is no need to disable DMP when Volume Manager is installed on the same host as Power Path, but it is a redundant solution.
Exclusive Use support requires Power Path version 1.5 or later.

Useful Commands

List the controllers on a specified enclosure or a particular type of enclosure

```
# vxdmpadm listctlr all
CTLR-NAME       ENCLR-TYPE      STATE           ENCLR-NAME
=========================================================
c2              Disk            ENABLED      Disk
c8              FC60            ENABLED      FC600
c10             FC60            ENABLED      FC600

# vxdmpadm listctlr enclosure=FC600 type=FC60
CTLR-NAME       ENCLR-TYPE      STATE           ENCLR-NAME
=========================================================
c8              FC60            ENABLED      FC600
c10             FC60            ENABLED      FC600
```

Obtain all paths connected to a particular controller

```
# vxdmpadm getsubpaths ctlr=c10
NAME         STATE          PATH-TYPE  DMPNODENAME  ENCLR-TYPE   ENCLR-NAME
=============================================================================
c10t0d0      ENABLED      PRIMARY    c8t0d0       FC60         FC600
c10t0d1      ENABLED      PRIMARY    c8t0d1       FC60         FC600
c10t0d2      ENABLED      PRIMARY    c8t0d2       FC60         FC600
c10t0d3      ENABLED      PRIMARY    c8t0d3       FC60         FC600
c10t0d4      ENABLED      PRIMARY    c8t0d4       FC60         FC600
c10t0d5      ENABLED      PRIMARY    c8t0d5       FC60         FC600
c10t0d6      ENABLED      PRIMARY    c8t0d6       FC60         FC600
c10t1d2      ENABLED      PRIMARY    c8t1d2       FC60         FC600
c10t1d3      ENABLED      PRIMARY    c8t1d3       FC60         FC600
```

Disabling and Enabling Controllers

DMP allows you to turn off I/O to a host I/O contoller so that you can perform administrative operations. This feature can be used for maintenance of controllers attached to the host or of disk arrays supported by VxVM. I/O operations to the host I/O controller can be turned back on after the mainanence task is completed.

To disable/enable IOs through the host disk controller c10, use the following commands:

```
# vxdmpadm disable ctlr=c10
# vxdmpadm enable ctlr=c10
```

List all paths controlled by the DMP node

```
# vxdmpadm getsubpaths dmpnodename=c8t0d0
NAME         STATE        PATH-TYPE  CTLR-NAME  ENCLR-TYPE   ENCLR-NAME
=============================================================================
c8t0d0       ENABLED      PRIMARY    c8         FC60         FC600
c10t0d0      ENABLED      PRIMARY    c10        FC60         FC600
```

If DMP is no working

1. Check the installed licenses

   ```
   # vxlicense –p
   ```

2. List device drivers in the system

```
# lsdev -C vxvm
    Character     Block       Driver          Class
       32           0         dmp             vxvm
       33           1         vol             vxvm
       36          -1         vols            vxvm
```

3. Is your Disk Array supported?

```
# vxddladm listsupport all
```

If an XP1024, XP128 or a VA7410 is connected to your system you need a special ASL (Array Support Library → Links in section "Additional information")

4. Check the serial numbers

```
#/etc/vx/diag.d/vxdmpinq /dev/rdsk/c7t12d0 | grep Serial
Serial Number          : 3EL01W85
#/etc/vx/diag.d/vxdmpinq /dev/rdsk/c6t12d0 | grep Serial
Serial Number          : 3EL01W85
#/etc/vx/diag.d/vxdmpinq /dev/rdsk/c6t13d0 | grep Serial
Serial Number          : 3EL0D6WX
#/etc/vx/diag.d/vxdmpinq /dev/rdsk/c7t13d0 | grep Serial
Serial Number          : 3EL0D6WX
```

5. You just added some new disks to one of your FC10s (example) and DMP is not working for the new disks:

```
#vxdmpadm listctlr all
CTLR-NAME       ENCLR-TYPE      STATE          ENCLR-NAME
=========================================================
c1              Disk            ENABLED      enc0
c2              Disk            ENABLED      enc0
c5              Disk            ENABLED      enc0
c6              Disk            ENABLED      enc0
c4              Disk            ENABLED      enc0
c7              Disk            ENABLED      enc0
c6              OTHER_DISKS     ENABLED      OTHER_DISKS ←
c7              OTHER_DISKS     ENABLED      OTHER_DISKS ←

#vxdmpadm getsubpaths ctlr=c6
NAME         STATE         PATH-TYPE  DMPNODENAME  ENCLR-TYPE   ENCLR-NAME
=============================================================================
[…]
c6t9d0       ENABLED         -        c6t9d0       Disk         enc0
c6t12d0      ENABLED         -        c6t12d0      OTHER_DISKS  OTHER_DISKS
c6t13d0      ENABLED         -        c6t13d0      OTHER_DISKS  OTHER_DISKS
c6t14d0      ENABLED         -        c6t14d0      OTHER_DISKS  OTHER_DISKS
c6t15d0      ENABLED         -        c6t15d0      OTHER_DISKS  OTHER_DISKS
```

→ Check the Vendor ID of the new disks

```
#/etc/vx/diag.d/vxdmpinq /dev/rdsk/c6t12d0

Inquiry for /dev/rdsk/c6t12d0, evpd 0x0, page code 0x0
Peripheral Qualifier/Device Type : 0
Removable bit                    : 0
Device type modifier             : 0
```

```
ISO Version                     : 0
ECMA Version                    : 0
ANSI Version                    : 3
Additional Length               : 8b
Vendor id               : HP 18.2G          ←
Product id              : ST318304FC
Revision Number         : HP12
Serial Number           : 3EL01W85
```

→ List all supported JBODs

```
#/usr/sbin/vxddladm listjbod

VID      PID              Opcode   Page Code   Page Offset      SNO length
==============================================================================
HITACHI  ALL PIDs            18         -1         36           12
SEAGATE  ALL PIDs            18         -1         36           12
```

"HP 18.2G" is not listed

→ Add a new entry for "HP 18.2G"

```
#/usr/sbin/vxddladm addjbod vid="HP 18.2G"
#/usr/sbin/vxddladm listjbod

VID      PID              Opcode   Page Code   Page Offset      SNO length
==============================================================================
HITACHI  ALL PIDs            18         -1         36           12
SEAGATE  ALL PIDs            18         -1         36           12
HP 18.2G ALL PIDs            18         -1         36           12
```

→ Cause vxconfigd to scan for any disks that were newly added since vxconfigd was last started.

```
#/usr/sbin/vxdctl enable
```

Commands Overview

Disk Operations	
Initialize disk	If LVM headers exist, `pvremove` them (may have to `pvcreate -f` first). `vxdiskadm` (good for setup and other things) or use `vxdiskadd c1t1d0`. `vxdisk list` (lists disks, specify disk to see disk header). To remove disk from VM control `vxdiskunsetup c1t1d0`.
List disks [header]	`vxdisk list [diskname]`
Disk Group Operations	
Create disk group	`vxdg init dg_name disk01=c1t1d0 disk02 ...`
List disk groups	`vxdg list ([dg_name]` for more info)
Add disk to group	`vxdg -g dg_name adddisk disk01=c1t1d0 disk02 ...`
Remove disk from group	`vxdg -g dg_name rmdisk disk01`
Deport disk group	`vxdg deport dg_name`
Import disk group	`vxdg import dg_name` (may then have to `vxrecover -g dg_name -sb`)
Destroy disk group	`vxdg destroy dg_name`
See free space	`vxassist -g dg_name maxsize [layout= ...]`
Volume Operations	
Create a volume	`vxassist -g dg_name make volname size layout=format disk01 disk02 ...`
Remove a volume	`vxedit -g dg_name [-rf] rm volname`
Rename a volume	`vxedit -g dg_name rename volname newvolname`
Resize a volume	`vxassist -g dg_name growto[by] volname newlength[lengthchange]` can also use `shrinkto[by]` or use `vxresize` to do the file system as well

Start/stop volumes	`vxvol -g dg_name start[stop] volname`
	`vxvol -g dg_name startall [stopall` (be careful)`]`
	`vxrecover -sn volname`

Move volume to another disk	`vxassist -g dg_name move volname !disk01 disk02`

Change layout:
add another stripe column
add mirror
remove mirror
	`vxassist -g dg_name relayout volname ncol=...`
	`vxassist -g dg_name mirror volname`
	`vxplex dis <plexname>` o
	`vxplex -o rm dis <plexname>` to remove plex as well

Snapshot a volume	`vxassist -g dg_name snapstart volname` `vxassist -g dg_name snapshot volname newvolname`
Add log to a volume	`vxassist -g dg_name addlog volname [alloc=disk..]`
Remove log	`vxplex -g dg_name -o rm dis volname`
Evacuate disk	`vxevac -g dg_name <fromdisk> <todisk>`
Plex Operations	
Create a plex	`vxmake -g dg_name plex plexname sd=subdiskname`

Associate a plex to a vol	`vxplex -g dg_name att volname plexname`
Dis-associate a plex	`vxplex dis plexname`
Remove a plex	`vxedit -g dg_name rm plexname`
List all plexes	`vxprint -lp`
Detach a plex	`vxplex -g dg_name det plexname`
Subdisk Operations	
Create a subdisk	`vxmake -g dg_name sd sdname disk,offset,length`
Remove a subdisk	`vxedit -g dg_name rm sdname`
Associate a subdisk to a vol	`vxsd assoc plexname sdname`
Dis-associate a subdisk	`vxsd dis sdname`
Join subdisk	`vxsd join sd1 sd2 newsdname`
Recovery	

Save config data to file
Look at dump
Restore config
	`vxprint (-g dg_name) -mvphsr > file`
	`cat file | vxprint -d - -ht`
	`vxmake -g dg_name -d file`
	`vxvol start volname`

Plex attach, raid5 subdisk recovery, resync	`vxrecover or vxplex att`
Start volume	`vxvol (-f) start volname`
Set plex states	`vxmend fix (clean|active|stale|empty) plex`
Initialise a volume	`vxvol init (active|clean|enable) volname`

Re-attaching a failed disk	`vxdctl enable`
	`vxdg -g dg_name -k adddisk faileddiskname=cntndn`
	`vxrecover -g dg_name -s faileddiskname`
	`/etc/vx/bin/vxbootsetup faileddiskname`

Non-VM boot disk failure where rootdg is on other disk(s)	Replace boot disk
	Install OS
	Install VxVM (don't do `vxinstall`)
	Add license keys
	`rm /etc/vx/reconfig.d/state.d/install-db`
	`vxiod set 10`
	`vxconfig -d` (ie. run in disabled mode)
	`vxdctl init hostname` (creates `/etc/vx/volboot`)
	`vxdctl enable`
	Reboot

Rootdg disk failure where rootdg has only the one disk but all other dg's are OK	`Replace boot disk`
	`Install OS`
	`Install VxVM`
	`vxinstall`
	`Reboot`

Dynamic Multipathing (DMP)	
Show all controllers	`vxdmpadm listctlr all`
Enable/disable controller	`vxdmpadm enable[disable] ctlr=……`
Stop & restart vxconfigd	`vxconfigd -k`
Disable/enable vxconfigd	`vxconfigd -k -m disable[enable]`
Turn on logging [to a file]	`vxconfigd -k [-x logfile=/tmp/log] -x 1 (up to 9)`
To redo vxinstall	`touch /etc/vx/reconfig.d/state.d/install-db`

Additional information

VERITAS Volume Manager 3.5 Administrator's Guide:
http://docs.hp.com/hpux/onlinedocs/5187-1369/5187-1369.html

VERITAS Volume Manager 3.5 Installation Guide:
http://docs.hp.com/hpux/onlinedocs/5187-1371/5187-1371.html

VERITAS Volume Manager 3.5 Migration Guide:
http://docs.hp.com/hpux/onlinedocs/5187-1372/5187-1372.html

VERITAS Volume Manager 3.2 Administrator's Guide:
http://docs.hp.com/hpux/onlinedocs/B7961-90025/B7961-90025.html

VERITAS Volume Manager 3.2 Migration Guide:
http://docs.hp.com/hpux/onlinedocs/B7961-90024/B7961-90024.html

WTEC JFS/VxFS Web Site:
http://wtec.cup.hp.com/~hpux/fs/vxvm.htm (HP internal)

Upgrade Scenarios:
http://wtec.cup.hp.com/~hpux/fs/VxVM/upgrade_process.htm (HP internal)

DMP and EMC Power Path Exclusive Use Support:
http://wtec.cup.hp.com/~hpux/fs/VxVM/vxvm_articles/dmp_and_emc_power_path_exclusive.htm
(HP internal)

VxVM contibuted Tools
http://wtec.cup.hp.com/~hpux/fs/VxVM/tools.htm (HP internal)

VERITAS Enabled Arrays - Array Support Library version 1.1 for HP StorageWorks disk arrays XP1024/128 (Active/Active) on HP-UX 11i
http://seer.support.veritas.com/docs/251522.htm (non HP)

VERITAS Enabled Arrays - Array Support Library for the HP StorageWorks Virtual Array 7410 on HPUX 11i
http://seer.support.veritas.com/docs/254689.htm (non HP)

Adding support for HP VA 7410 disk arrays as JBOD will make disk groups and volumes disappear.
http://seer.support.veritas.com/docs/253417.htm (non HP)