

Chapter 17

-

HP VERITAS Volume Manager (VxVM)

INDEX

Introduction	3
Features	3
Functionality of the Base HP VERITAS Volume Manager	3
Functionality of the HP VERITAS Volume Manager 3.1	4
Base versus Full Version.....	5
swlist Output and Licencing	6
Confirm that the Product is Licensed.....	6
Architecture	7
VxVM Objects	8
VMSA - Volume Manager Storage Administrator	10
Confirm that the vmsa server is not running	10
Start the vmsa server	10
Stop the vmsa server	10
Auto-Start Activation	10
Confirm that Auto-Start is enabled	10
VMSA directories	10
Disk Group Tasks	11
Disk Maintenance	15
List VM-Disks.....	15
List disk details	15
Status of a disk	15
Volume Maintenance	16
Internals	24
How to read the header information of a disk.....	24
Recovery	25
Debug level for vxconfigd	25
/etc/vx/volboot.....	25
Disk failure / Disk replacement.....	26
Reattaching Disks.....	27
Recovering the Volume Manager Configuration.....	27
How to recover rootdg that had only 1 disk after a disk failure.....	29
Dynamic Multipathing (DMP)	30
Questions & Answers.....	30
New features of VM 3.11 summary for DMP.....	32
Files and Directories	33
Commands Overview	35
Additional information	38

Introduction

VERITAS Volume Manager 3.1 for HP-UX provides state-of-the-art online disk management for HP-UX. VxVM consists of several related products:

Base VERITAS Volume Manager 3.1 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 3.1 for HP-UX also provides some basic clustering features.

VERITAS Volume Manager 3.1 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 3.1 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 3.1 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.

Features

Functionality of the Base HP VERITAS Volume Manager

- Java-based admin GUI
The java-based admin GUI (VMSA), provides functionalities like disk, diskgroup and volume maintenance in an easy graphical way. Alternative Interfaces are the Command Line Interface (CLI) and a semiinteractive Interface (vxdiskadm).
- Striping (RAID 0)
- Concatenation
Concatenation is the combination of discontiguous regionen to a virtual device.
- Path failover support (active/passive peripherals)
Dynamic Multipathing (DMP) is for mass-storage products with multiple Controllers. "active/passive peripherals" run I/O-Requests only about one I/O-Path. In case of I/O path failure, DMP will switch automatically to to a redundant I/O Path.
- Online Resizing of Volumes
The online resizing is already with Base HP VERITAS Volume Manager version

possible.

- Task monitor
You can monitor specific tasks and you can control the speed while volume recovery processes (only single systems) are running.

Functionality of the HP VERITAS Volume Manager 3.1

- Load balancing--DMP (active/active peripherals)
With help of Dynamic Multipathing (DMP) run I/O-Requests parallel over both I/O-paths.
- Hot relocation/unrelocation
VxVM monitors Disk Failure Events, informs the administrator via email, will find the concerning area, moves the data to a free functional area within the diskgroup. Sufficient space for redundancy is required.
Unrelocation means removing of the data to the original area after troubleshooting.
- The maximum number of mirrors is 32
- Different Layouts
Mirroring (RAID-1)
Mirrored Stripes (RAID 1+0)
Striped Mirrors (RAID 0+1)
RAID-5
- Online Migration
Online Data Migration supports the dynamic movement of data area from one disk to another disk and vice versa
- Online Relayout
Layout changes, e.g. Concatenated to RAID-5, without interrupting the online data transfer.

Base versus Full Version

Base HP VERITAS Volume Manager: B7961AA	HP VERITAS Volume Manager: B9116A
<p style="text-align: center;">Java based admin GUI</p> <p style="text-align: center;">RAID 0 (Striping)</p> <p style="text-align: center;">Concatentation (combining of discontiguous disk regions into virtual devices)</p> <p style="text-align: center;">Online resizing of volumes</p> <p style="text-align: center;">Task monitoring and speed control during volume recovery (single system VxVM only)</p>	
	<p>Load balancing DMP (Dynamic Multi Pathing)</p> <p>Hot-Relocation / Unrelocation</p> <p>RAID 1 (Mirroring) RAID 1+0 (Mirrored Stripes) RAID 0+1 (Striped Mirrors) RAID 5 (Requires additional License) supports 32 mirrors</p> <p>Online Data Migration</p> <p>Online Re-Layout</p>

swlist Output and Licencing

```
# swlist -l product B7961AA B9116AA
# Initializing...
# Contacting target "hprtd32"...
#
# Target:  hprtd32:/
#
# B7961AA          B.03.10.5      Base HP VERITAS Volume Manager 3.1
  B7961AA.HPvxvm   B.03.10.5      Base HP VERITAS Volume Manager 3.1
  B7961AA.HPvmsa   B.03.10.5      HP VERITAS Volume Manager 3.1 Storage
Administrator
# B9116AA          B.03.10.5      HP VERITAS Volume Manager 3.1
  B9116AA.VxVM-LIC B.03.10.5      HP VERITAS Volume Manager 3.1
```

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 Jul 15 10:00:00 2007 (2383.7 days from now)
vrts:vxlicense: INFO: Release Level: 25
vrts:vxlicense: INFO: Machine Class: All
vrts:vxlicense: INFO: Site ID: 0

# vxlicense -p (shows all licenses)

vrts:vxlicense: INFO: Feature name: HP_OnlineJFS [50]
vrts:vxlicense: INFO: Number of licenses: 1 (non-floating)
vrts:vxlicense: INFO: Expiration date: Sun Jun 24 10:00:00 2007 (2216.9 days from now)
vrts:vxlicense: INFO: Release Level: 22
vrts:vxlicense: INFO: Machine Class: All
vrts:vxlicense: INFO: Site ID: 0

vrts:vxlicense: INFO: Feature name: VxVM [95]
vrts:vxlicense: INFO: Number of licenses: 1 (non-floating)
vrts:vxlicense: INFO: Expiration date: Sun Jul 15 10:00:00 2007 (2237.9 days from now)
vrts:vxlicense: INFO: Release Level: 25
vrts:vxlicense: INFO: Machine Class: All
vrts:vxlicense: INFO: Site ID: 0
```

see also

```
# ll /etc/vx/elm/*.lic
-rw-r--r--  1 root      root      261 May 28 14:25 /etc/vx/elm/50.lic
-rw-r--r--  1 root      root      261 May 28 14:25 /etc/vx/elm/95.lic
```

Architecture

vxconfigd – vxdctl

The `vxconfigd` daemon is the interface between the VxVM commands and the kernel device drivers. The `vxdctl` command controls `vxconfigd` and does the following tasks:

- Manages boot information and VxVM root configuration initialization.
- The `vxconfigd` (`/dev/vx/config`) device is the interface through which all changes to the volume driver state are performed. This device can be opened by only one process at a time.
- Manipulates the contents of the `/etc/vx/volboot` file; the file contains a list of disks containing root configuration databases.
- The configuration database is kept in the private regions of the disk, in the kernel and in the `/dev/vx/tempdb`. There are also replicated Volume Manager configurations within a diskgroup.
- `vxconfigd` controls licensing.

If Dynamic Multipathing (DMP) is used, `vxdctl` also does the following:

- Reconfigures the DMP database when disk devices are added and removed from the system.
- Creates DMP device files in the device directories, `/dev/vx/dmp` and `/dev/vx/rdmp`.
- Reflects changes in path type for active/passive disk arrays. You can change the path type from primary to secondary and vice-versa through the utilities provided by disk array vendors.

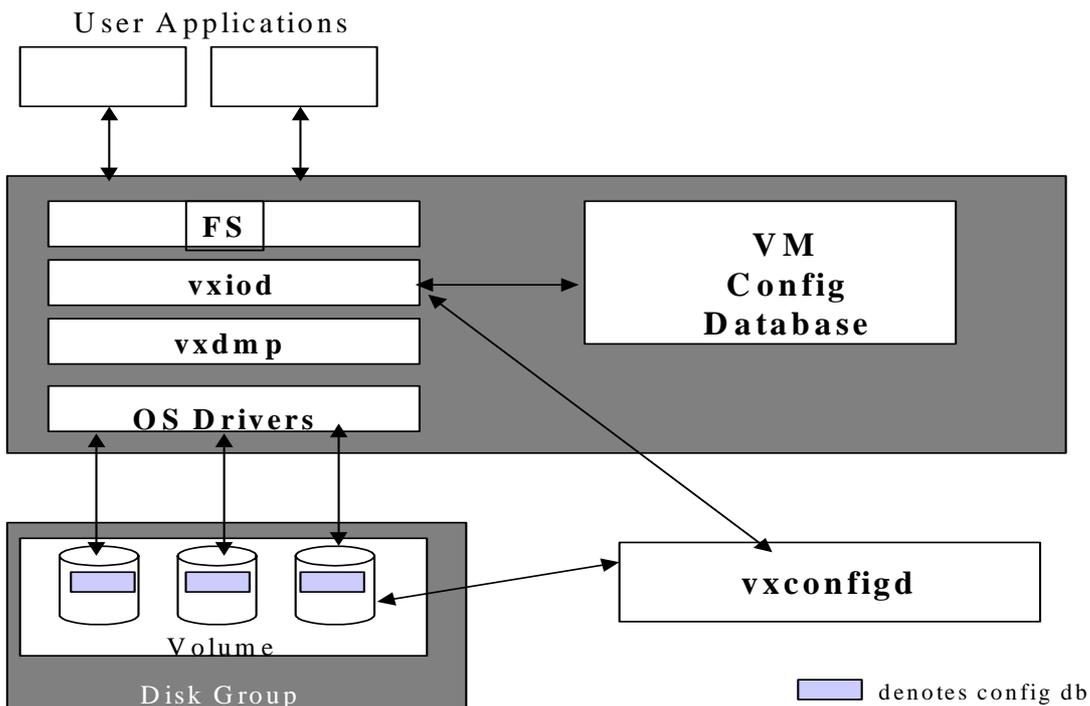
You have got the possibility to configure the `vxconfigd` in debug mode (see man page)

vxrelocd (hotrelocation / unrelocation); vxnotify

The `vxrelocd` monitors failed disk events, informs the administrator via email (`vxnotify`) and will find the concerning area within a diskgroup. It will move that data to a non failed area. You need free spare (e.g. recommend are spare disks for performance reasons, free space within the diskgroup is also possible) for this action.

vxiod

Extended I/O daemon for extended I/O operations without blocking calling processes.



VxVM Objects

Disk Groups (e.g. roodg)

A collection of VM-Disks. A disk group has one or more volumes.

Disk drives can be shared by two or more hosts, but accessed by only one host at a time. If one host 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)

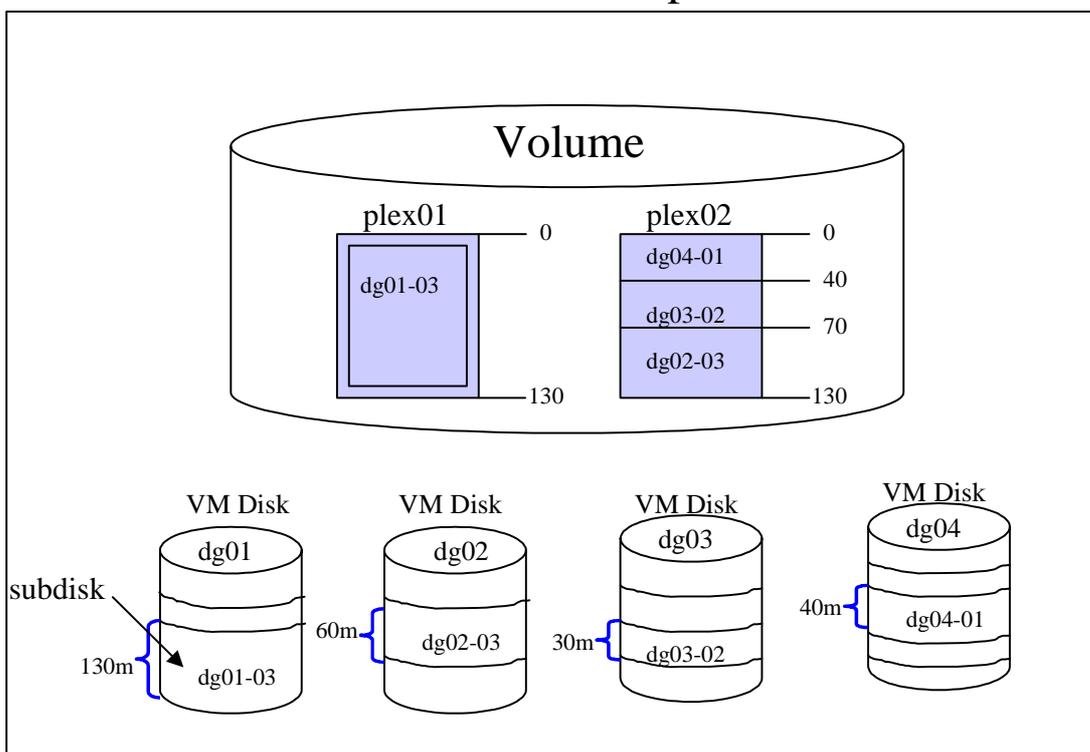
VM-Disks are physical disk that are under VxVM control and contain one or more subdisks. The VM-Disks consist of private and public region:

private region	public region
<ol style="list-style-type: none"> disk header <ul style="list-style-type: none"> label diskgroup info: <ul style="list-style-type: none"> disk media name (disk-0X) diskid disk access name (c#t#d#) hostid pointers to the private and public regions configuration database <ul style="list-style-type: none"> contains the VxVM objects kernel logs <ul style="list-style-type: none"> contains configuration changes (log plexes attached, object creation / deletion) 	<p>The public region is used for allocating subdisk to use it for:</p> <p>data storage</p>

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.

Disk Group



VMSA - Volume Manager Storage Administrator

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/
-rw-r--r--  1 root    root
-rw-r--r--  1 root    sys
-rw-r--r--  1 root    sys
-rw-r--r--  1 root    sys
```

Logfiles

```
18 May  9 10:38 .server_pids
3514 May  9 10:38 access
1128 May  9 10:57 command
5872 May  9 10:38 server.log
```

```
# ll /opt/HPvmsa/bin/
```

```
-rwxr-xr-x  1 bin     bin
-rwxr-xr-x  1 bin     bin
-rwxr-xr-x  1 bin     bin
```

Commands for start/stop

```
6308 Sep 18  2000 autostart
2562 Sep 18  2000 vmsa
7635 Sep 18  2000 vmsa_server
```

```
# vi /opt/HPvmsa/vmsa/properties
```

Customization of the GUI properties

Disk Group Tasks

Diskgroup specifical listings

```
# vxdg free
GROUP      DISK      DEVICE    TAG      OFFSET  LENGTH  FLAGS
rootdg     disk01    c6t0d0    c6t0d0    0       17782088 -
rootdg     disk02    c6t8d0    c6t8d0    0       17782088 -
peterdg    peterdg01 c6t1d0    c6t1d0    0       17782088 -

# vxdg list
NAME      STATE      ID
rootdg    enabled    978536140.1025.hprrdd32
peterdg   enabled    979069695.1075.hprrdd32

# vxdg list rootdg
Group:     rootdg
dgid:     978536140.1025.hprrdd32
import-id: 0.1
flags:
version:   70
local-activation: read-write
detach-policy: local
copies:    nconfig=default nlog=default
config:    seqno=0.1076 permlen=727 free=725 templen=2 loglen=110
config disk c6t0d0 copy 1 len=727 state=clean online
config disk c6t8d0 copy 1 len=727 state=clean online
log disk c6t0d0 copy 1 len=110
log disk c6t8d0 copy 1 len=110
```

Intitalize a disk/uninitialized disk → free disk pool

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

- creates private and public regions on the disk
- does not put the disk in the diskgroups

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.

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

```
# /etc/vx/bin/vxdiskunsetup -C c6t1d0
vxdiskunsetup: c6t1d0: Disk device is in use
```

Adding a disk

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

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

Create a new diskgroup (the disk should be initialized)

```
# vxdg init <dg_name> <dm_name_old>=<dm_name_new>

# vxdg init peter peter01=c6t1d0
peter    = dg_name
peter01  = Diskmedianame (dm_name)
c6t1d0   = Diskaccessname(da_name)

# vxdg list peter
Group:    peter
dgid:    979073966.1078.hprrdd32
import-id: 0.1077
flags:
version:  70
local-activation: read-write
detach-policy: local
copies:   nconfig=default nlog=default
config:   seqno=0.1027 permlen=727 free=726 templen=1 loglen=110
config disk c6t1d0 copy 1 len=727 state=clean online
log disk c6t1d0 copy 1 len=110
```

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

```
# vxdg -g rootdg adddisk disk02=c6t8d0
disk02 = Diskmedianame (DM)
c6t8d0 = Diskaccessname(DA)

# vxdisk list
DEVICE      TYPE      DISK      GROUP      STATUS
c6t0d0      simple    disk01    rootdg     online
c6t1d0      simple    peter01   peter      online
c6t8d0      simple    disk02    rootdg     online
```

Add further disks:

```
# vxdg -g rootdg adddisk disk02=c6t8d0 disk03=cXtYdZ ..
disk0X = Diskmedianame (dm_name)
cXtYdZ = Diskaccessname(da_name)
```

Rename a disk in a disk group

```
# 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 rootdg rmdisk disk02
disk02 = Diskmedianame (dm_name)

# vxdisk list
DEVICE      TYPE      DISK      GROUP      STATUS
c6t0d0      simple    disk01    rootdg     online
c6t1d0      simple    peter01   peter      online
c6t8d0      simple    -         -         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 peter peter01 peter02
peter    = dg_name
peter01, peter02 = Diskmedianame (dm_name)
```

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

Destroy the last disk of a disk group

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

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

- host ID
- diskname
- dg_name

In case, that the disk group was accidentally removed, you can only reimport via the CLI (not vxdiskadm and VMSA), by using the group ID.

Deport of a disk group

```
# vxdg deport <dg_name>
```

```
# vxdg deport peter
```

```
before the deport: # vxdisk list
```

```
c6t0d0      simple   disk01      rootdg      online
c6t1d0      simple   disk01      peter       online
c6t8d0      simple   disk02      peter       online
```

```
after the deport: # vxdisk list
```

```
c6t0d0      simple   disk01      rootdg      online
c6t1d0      simple   -           -           online
c6t8d0      simple   -           -           online
```

```
Disk:  c6t1d0
type:  simple
flags: online ready private autoconfig
diskid: 979160280.1082.hprtdd32
dname: peter
dgid:  979160356.1085.hprtdd32
hostid:
info:  privoffset=128
```

```
Disk:  c6t8d0
type:  simple
flags: online ready private
diskid: 978536149.1060.hprtdd32
dname: peter
dgid:  979160356.1085.hprtdd32
hostid:
info:  privoffset=128
```

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 diskgroupname

```
# vxdg -n <dg_name_new> deport <dg_name_old>
```

Deport to a specific hostname and a specific diskgroupname

```
# vxdbg -n <dg_name_new> -h <hostname_name_new> deport <dg_name_old>
```

Display deported diskgroups

```
# vxdisk -o alldgs list
```

DEVICE	TYPE	DISK	GROUP	STATUS
c6t0d0	simple	disk01	rootdg	online
c6t1d0	simple	-	(peter)	online
c6t8d0	simple	-	(peter)	online

```
# vxdisk -s list c6t1d0
```

```
Disk: c6t1d0
type: simple
flags: online ready private autoconfig
diskid: 979160280.1082.hprrdd32
dgname: peter
dgid: 979160356.1085.hprrdd32
hostid:
info: privoffset=128
```

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

Import of disk groups

```
# vxdbg import <dg_name>
```

Import of a disk group and rename the disk group name

```
# vxdbg -n <dg_name_new> import <dg_name_old>
```

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

```
# vxdbg -t -n <dg_name_new_temp> import <dg_name_old>
```

Import of a disk group and clear the import locks

```
# vxdbg -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`)

```
# vxdbg -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
c1t0d0      simple   -         -          LVM
c2t0d0      simple   -         -          LVM
c4t0d0      simple   -         -          LVM
c4t1d0      simple   -         -          LVM
c4t8d0      simple   -         -          LVM
c4t9d0      simple   -         -          LVM
c6t0d0      simple   disk01    rootdg     online
c6t1d0      simple   peterdg01 peterdg     online
c6t8d0      simple   disk02    rootdg     online
```

Comment: the `vxdisk -s list` → gives a Summary of Listing and more details

List disk details

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

```
# vxdisk list disk01
```

```
Device:      c6t0d0
devicetag:   c6t0d0
type:        simple
hostid:      hprtdd32
disk:        name=disk01 id=978536141.1039.hprtdd32
timeout:     30
group:       name=rootdg id=978536140.1025.hprtdd32
info:        privoffset=128
flags:       online ready private autoimport imported
pubpaths:    block=/dev/vx/dmp/c6t0d0 char=/dev/vx/rdmp/c6t0d0
version:     2.1
iosize:      min=1024 (bytes) max=64 (blocks)
public:      slice=0 offset=1152 len=17782088
private:     slice=0 offset=128 len=1024
update:      time=978691625 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:    2
c6t0d0      state=enabled
c7t0d0      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 peter
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      USETYPE     PREFPLEX    RDPOL
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 peter      default      default    2635000    979160356.1085.hprrtd32
dm disk01     c6t1d0      simple     1024       17782088    -
dm disk02     c6t8d0      simple     1024       17782088    -
sd disk01-02  volume2-01  disk01     358400     204800      1/0         c6t1d0     ENA
sd disk02-01  volume2-01  disk02     0           204800      0/0         c6t8d0     ENA
pl volume2-01 volume2      ENABLED    ACTIVE     409600      STRIPE      2/128      RW
v volume2     -           ENABLED    ACTIVE     409600      fsgen       volume2-01 SELECT
```

```
type          Number of column/explanation
```

Disk Media

1. Record type (**dm**)
2. Record name
3. Underlying disk access record
4. Disk access record type (simple or nopriv)
5. Length of the disk's private region
6. Length of the disk's public region
7. Path to use for accessing the underlying raw disk device for the disk's public region.

Subdisk

1. Record type (**sd**)
2. Record name
3. Associated plex, or dash (-) if the subdisk is dissociated
4. Name of the disk media record used by the subdisk
5. Device offset in sectors
6. Subdisk length in sectors
7. Plex association offset, optionally preceded by subdisk column number for subdisks associated to striped plexes, LOG for log subdisks, or the putil[0] field if the subdisk is dissociated. The putil[0] field can be non-empty to reserve the subdisk's space for non-volume uses. If the putil[0] field is empty, - is displayed for dissociated subdisks.
8. A string representing the state of the subdisk (ENA if the subdisk is usable; DIS if the subdisk is disabled; RCOV if the subdisk is part of a RAID-5 plex and has stale content; DET if the subdisk has been detached; KDET if the subdisk has been detached in the kernel due to an error; RMOV if the media record on which the subdisk is defined has been removed from its disk access record by a utility; RLOC if a subdisk has failed and is waiting to be relocated, or NDEV if the media record on which the subdisk is defined has no access record associated).

subvolumes

1. Record type (**sv**)
2. Record name
3. Associated plex, or dash (-) if the subvolume is dissociated.

	<ol style="list-style-type: none"> 4. Name of the underlying (layered) volume record used by the subvolume. 5. Number of layers used in the subvolume. 6. Subvolume length in sectors 7. Plex association offset, optionally preceded by subvolume column number for subvolumes associated to striped plexes. 8. Number of active plexes, followed by the number of plexes in the underlying (layered) volume. 9. A string representing the state of the subvolume (ENA if the subvolume is usable; DIS if the subvolume is disabled; KDET if the subvolume has been detached in the kernel due to an error
plexes	<ol style="list-style-type: none"> 1. Record type (pl) 2. Record name 3. Associated volume, or - if the plex is dissociated 4. Plex kernel state 5. Plex utility state. If an exception condition is recognized on the plex (an I/O failure, a removed or inaccessible disk, or an unrecovered stale data condition), then that condition is listed instead of the value of the plex record's state field. 6. Plex length in sectors 7. Plex layout type 8. Number of columns and plex stripe width, or - if the plex is not striped 9. Plex I/O mode, either RW (read-write), WO (write-only), or RO (read-only)
volumes	<ol style="list-style-type: none"> 1. Record type (v) 2. Record name 3. Associated usage type 4. Volume kernel state 5. Volume utility state 6. Volume length in sectors 7. Volume read policy 8. The preferred plex, if the read-policy uses a preferred plex

see also the manpage (and grep for "Record Type")

Volume Read Policies

Read Policies with mirroring

- *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 preferred 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.


```
# vxprint -g rootdg | grep 3-stripe-volume
TY NAME          ASSOC          KSTATE  LENGTH  PLOFFS  STATE  TUTILO  PUTILO
dg rootdg        rootdg         -        -        -        -        -        -

dm disk01        c6t0d0         -        17782088 -        -        -        -
dm disk02        c6t1d0         -        17782088 -        -        -        -
dm disk03        c6t8d0         -        17782088 -        -        -        -

v 3-striped-volume fsgen      ENABLED  51200   -        ACTIVE  -        -
pl 3-striped-volume-01 3-striped-volume ENABLED  51264 - ACTIVE  -        -
sd disk01-02     3-striped-volume-01 ENABLED  17088  0        -        -        -
sd disk02-02     3-striped-volume-01 ENABLED  17088  0        -        -        -
sd disk03-02     3-striped-volume-01 ENABLED  17088  0        -        -        -

# vxassist -g rootdg make 3-striped-volume+log 50m layout=stripe,log

# vxprint -g rootdg | grep 3-stripe-volume+log
v 3-striped-volume+log fsgen ENABLED  51200   -        ACTIVE  -        -
pl 3-striped-volume+log-01 3-striped-volume+log ENABLED  51264 - ACTIVE  -        -
sd disk01-03     3-striped-volume+log-01 ENABLED  17088  0        -        -        -
sd disk02-03     3-striped-volume+log-01 ENABLED  17088  0        -        -        -
sd disk03-03     3-striped-volume+log-01 ENABLED  17088  0        -        -        -
pl 3-striped-volume+log-02 3-striped-volume+log ENABLED  LOGONLY - ACTIVE  -        -
sd disk01-04     3-striped-volume+log-02 ENABLED  33 LOG   -        -        -
```

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)

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

Example: rootdg with 3 disks

```
# vxassist -g rootdg make raid5-volume 200m layout=raid5
vxvm:vxassist: ERROR: Too few disks for striping; at least 4 disks are needed
```

```
# vxassist -g rootdg make raid5-volume 200m layout=raid5 ncols=3
vxvm:vxassist: ERROR: Cannot allocate space for 204800 block volume
```

```
# vxassist -g rootdg make raid5-volume 50m layout=raid5 ncols=3
vxvm:vxassist: ERROR: Cannot allocate space for 51200 block volume
```

```
# vxassist -g rootdg make raid5-volume 200m layout=raid5,nolog
this works, because the minimum numbers without logplex is 3
```

```
# vxprint -tg rootdg
```

```
DG NAME          NCONFIG      NLOG      MINORS  GROUP-ID
DM NAME          DEVICE       TYPE      PRIVLEN PUBLLEN  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   USETYPE    PREFPLEX  RDPOL

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 rootdg          default      default  0          978536140.1025.hprtd32
dm disk01          c6t0d0      simple  1024      17782088 -
dm disk02          c6t1d0      simple  1024      17782088 -
dm disk03          c6t8d0      simple  1024      17782088 -
sd disk01-01      raid5-volume-01 disk01  0          102400    0/0        c6t0d0    ENA
sd disk02-01      raid5-volume-01 disk02  0          102400    1/0        c6t1d0    ENA
sd disk03-01      raid5-volume-01 disk03  0          102400    2/0        c6t8d0    ENA
pl raid5-volume-01 raid5-volume  ENABLED  ACTIVE    204800    RAID       3/16      RW
v  raid5-volume -          ENABLED  ACTIVE    204800    raid5      -         RAID

```

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> mirror=2
```

Example:

```

# vxassist -g peter make volume1 100m layout=mirror

# vxprint -tg peter
dg peter          default      default  2635000   979160356.1085.hprtd32
dm disk01          c6t1d0      simple  1024      17782088 -
dm disk02          c6t8d0      simple  1024      17782088 -
sd disk01-01      volume1-01  disk01  0          102400    0          c6t1d0    ENA
sd disk02-01      volume1-02  disk02  0          102400    0          c6t8d0    ENA
pl volume1-01    volume1     ENABLED  ACTIVE    102400    CONCAT     -         RW
pl volume1-02    volume1     ENABLED  ACTIVE    102400    CONCAT     -         RW
v  volume1       -          ENABLED  ACTIVE    102400    fsgen     -         SELECT

```

Mirroring an existing volume

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

```
# vxassist mirror volume_name
```

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

```
# vxassist mirror voltest
```

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

```

# vxmake plex plex_name sd= subdisk_name ...
# vxplex att volume_name plex_name

```

Create a concat-mirror layout

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

Example:

```
# vxassist -g peter make mirror-concat 50m layout=mirror-concat
```

```
Disk group: peter
```

```

TY NAME          ASSOC          KSTATE    LENGTH    PLOFFS    STATE      TUTILO    PUTILO
dg peter         peter         -         -         -         -         -         -
dm disk01        c4t8d0        -         17782088 -         -         -         -
dm disk02        c4t9d0        -         17782088 -         -         -         -
dm disk03        c6t1d0        -         17782088 -         -         -         -
dm disk04        c6t8d0        -         17782088 -         -         -         -

v mirror-concat fsgen          ENABLED   10240    -         ACTIVE   -         -
pl mirror-concat-01 mirror-concat ENABLED   10240    -         ACTIVE   -         -
sd disk01-01     mirror-concat-01 ENABLED   10240    0         -         -         -
pl mirror-concat-02 mirror-concat-02 ENABLED   10240    -         ACTIVE   -         -
sd disk02-01     mirror-concat-02 ENABLED   10240    0         -         -         -

```

Create a concatenated-pro layout (concat-mirror)

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

Example:

```

# vxassist -g peter make concatenated-pro 50m layout=concat-mirror

dg peter         default          default  1133000  989502281.1506.hprtd32
dm disk01        c4t8d0          simple   1024     17782088 -
dm disk02        c4t9d0          simple   1024     17782088 -
dm disk03        c6t1d0          simple   1024     17782088 -
dm disk04        c6t8d0          simple   1024     17782088 -

v concatenated-pro -          ENABLED  ACTIVE  51200    fsgen    -          SELECT
pl concatenated-pro-03 concatenated-pro ENABLED  ACTIVE  51200    CONCAT  -          RW
sv concatenated-pro-S01 concatenated-pro-03 concatenated-pro-L01 1 51200 0 2/2 ENA
v2 concatenated-pro-L01 -          ENABLED  ACTIVE  51200    fsgen    -          SELECT
p2 concatenated-pro-P01 concatenated-pro-L01 ENABLED  ACTIVE  51200    CONCAT  -          RW
s2 disk01-02     concatenated-pro-P01 disk01 0 51200    0        c4t8d0  ENA
p2 concatenated-pro-P02 concatenated-pro-L01 ENABLED  ACTIVE  51200    CONCAT  -          RW
s2 disk02-02     concatenated-pro-P02 disk02 0 51200    0        c4t9d0  ENA

```

Create striped-mirrored layout

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

Example:

```

# vxassist -g peter make concatenated-pro 50m layout=concat-mirror

dg peter         default          default  1133000  989502281.1506.hprtd32

dm disk01        c4t8d0          simple   1024     17782088 -
dm disk02        c4t9d0          simple   1024     17782088 -
dm disk03        c6t1d0          simple   1024     17782088 -
dm disk04        c6t8d0          simple   1024     17782088 -

v striped-pro    -          ENABLED  ACTIVE  51200    fsgen    striped-pro-03
SELECT
pl striped-pro-03 striped-pro    ENABLED  ACTIVE  51200    STRIPE   2/64     RW
sv striped-pro-S01 striped-pro-03 striped-pro-L01 1 25600 0/0 2/2 ENA
v2 striped-pro-L01 -          ENABLED  ACTIVE  25600    fsgen    -          SELECT
p2 striped-pro-P01 striped-pro-L01 ENABLED  ACTIVE  25600    CONCAT  -          RW
s2 disk01-02     striped-pro-P01 disk01 0 25600    0        c4t8d0  ENA
p2 striped-pro-P02 striped-pro-L01 ENABLED  ACTIVE  25600    CONCAT  -          RW
s2 disk03-02     striped-pro-P02 disk03 0 25600    0        c6t1d0  ENA
sv striped-pro-S02 striped-pro-03 striped-pro-L02 1 25600 1/0 2/2 ENA
v2 striped-pro-L02 -          ENABLED  ACTIVE  25600    fsgen    -          SELECT
p2 striped-pro-P03 striped-pro-L02 ENABLED  ACTIVE  25600    CONCAT  -          RW
s2 disk02-02     striped-pro-P03 disk02 0 25600    0        c4t9d0  ENA

```

```
p2 striped-pro-P04 striped-pro-L02 ENABLED ACTIVE 25600 CONCAT - RW
s2 disk04-02 striped-pro-P04 disk04 0 25600 0 c6t8d0 ENA
```

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 rm volume_name
# vxedit -rf rm volume_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 remove volume volume_name
```

Devices for the generated Volumes

```
# /dev/vx/rdisk/dg_name/vol_name
```

Example:

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

# newfs -F vxfs /dev/vx/rdsk/peter/volume1
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 set mode=rwx <vol_name>
```

Example:

```
# vxedit set mode=666 <vol_name>
```

Check the permissions of the volume

```
# vxprint -l <vol_name> | grep perms:
```

```
# vxedit set mode=666 volume1
# vxprint -l volume1 | grep perms:
perms:      user=root group=root mode=0666
```

Check the read policy

```
# vxprint -l <vol_name>
```

```
# vxprint -l volume1 | 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:

```
# vxvol rdpol round volume1
# vxprint -l volume1 | 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>
```

Example:

```
# vxedit set comment="first plex" volume1-01
# vxprint -l volume1-01
Disk group: peter
Plex:      volume1-01
info:     len=102400
type:     layout=CONCAT
state:    state=ACTIVE kernel=ENABLED io=read-write
assoc:    vol=volume1 sd=disk01-01
flags:    busy complete
comment:  first plex
```

Internals

How to read the header information of a disk

xd - command

Example:

```
# xd -tc -j128k -N8 /dev/rdisk/clt4d0
0000000  P R I V H E A D

# xd -tc -j131116 -N24 /dev/rdisk/clt4d0
0000000  1 0 0 3 9 1 9 9 5 2 . 1 0 4 0 .
0000010  g r c d g 4 5 5

# xd -tc -j131244 -N24 /dev/rdisk/clt4d0
0000000  1 0 0 3 9 1 9 9 4 9 . 1 0 2 5 .
0000010  g r c d g 4 5 5
```

compared with the

```
# vxdisk list clt4d0 | grep -e group: -e disk:
disk:      name=disk01 id=1003919952.1040.grcdg455
group:     name=rootdg id=1003919949.1025.grcdg455
```

vxprivutil

Example:

```
# /usr/lib/vxvm/diag.d/vxprivutil scan /dev/dsk/clt4d0
diskid: 1003919952.1040.grcdg455
group:  name=rootdg id=1003919949.1025.grcdg455
flags:  private autoimport
hostid: grcdg455
version: 2.1
iosize: 1024
public: slice=0 offset=1152 len=4101723
private: slice=0 offset=128 len=1024
update: time: 1003919961 seqno: 0.7
headers: 0 248
configs: count=1 len=727
logs:    count=1 len=110
```

dd - command

Example:

```
# dd if=/dev/rdisk/c4t8d0 bs=1k skip=128 count=1 | strings
1+0 records in
1+0 records out
PRIVHEAD
982746255.1258.hprrdd32
```

```
# /usr/lib/vxvm/diag.d/vxprivutil scan /dev/rdisk/c4t8d0
diskid: 982746255.1258.hprtd32
group: name= id=
flags: private autoimport
hostid:
version: 2.1
iosize: 1024
public: slice=0 offset=1152 len=17782088
private: slice=0 offset=128 len=1024
update: time: 982746849 seqno: 0.6
headers: 0 248
configs: count=1 len=727
logs: count=1 len=110
```

Recovery

Debug level for vxconfigd

Example:

```
# vxconfigd -x <debuglevel> <debuglevel> = 1-9
```

If direct logging is enabled, the default log file is `/var/adm/vxconfigd.log`. Or you can log all debug and error messages to the specified log file:

```
# vxconfig -x logfile=<file> -x <debuglevel>
```

/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.

```
# vxdctl list
Volboot file
version: 3/1
seqno: 0.1
cluster protocol version: 20
hostid: hprtd32 → Owner
entries:
```

vxdctl init creates the file `/etc/vx/volboot`

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
mktng-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 `mktng-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

1. 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)
```

2. Replacing a disk

If a disk was replaced due to a disk failure and you wish to move hot-relocate subdisks back to this replaced disk, see Chapter 6 , Administratotor 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/B7961-90002/B7961-90002.html>

How to recover rootdg that had only 1 disk after a disk failure

(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 replacement disk are both `/dev/rdisk/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 systems 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.

Disabling and Enabling Controllers

To disable/enable IOs through the host disk controller c2, use the following commands:

```
# vxdmpadm disable ctlr=c2
# vxdmpadm enable ctlr=c2
```

List all paths controlled by the DMP node

```
# vxdmpadm getsubpaths dmpnodename=c2t1d0
```

Obtain all paths connected to a particular controller

```
# vxdmpadm getsubpaths ctrl=c2
```

Determine the node that controls a path to a disk array

```
# vxdmpadm getdmpnode nodename=c3t2d1
```

Assign a new name to a disk array using the following command

```
# vxdmpadm setattr enclosure nike0 name=VMGRP_1
```

Display the DMP nodes to the enclosure

```
# vxdmpadm getdmpnode enclosure=nike0
```

List all paths controlled by the DMP node

```
# vxdmpadm getsubpaths dmpnodename=c2t1d0
```

Obtain all paths through a the host disk controller

```
# vxdmpadm getsubpaths ctrl=c2
```

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

```
# vxdmpadm listctrlr all  
# vxdmpadm listctrlr enclosure=nikee0 type=NIKE
```

Display all attributes of the enclosure(s)

```
# vxdmpadm listenclosure nike0 fc1010d0
```

DMP Restore Daemon

The DMP restore daemon periodically analyzes the health of the paths. The restore daemon is started at the system startup time with default attributes.

The interval of polling can be set by using the following command:

```
# vxdmpadm start restore interval=400
```

Set the policy check

Related to the certain policy it will check the health of paths that were previously disabled or checks all paths. The paths that are back online are revived and the inaccessible paths are disabled.

```
# vxdmpadm start restore policy=check_all
```

To stop the restore daemon enter the command

```
# vxdmpadm stop restore
```

New features of VM 3.11 summary for DMP

<http://seer.support.veritas.com/docs/235981.htm> (non HP)

The method to disable multipathing to devices in VxVM 3.1 and prior releases was to completely remove the VxDMP layer (i.e not to have the vxdmp module compiled into the kernel for HPUX). This methodology is no longer supported in VxVM 3.1.1 and later releases of Volume Manager.

The typical scenarios in which the vxdmp module was disabled are:

- existence of certain other multipathing drivers on the system with which VxDMP didn't coexist.
- certain devices are not handled properly by VxDMP.

In these scenarios adopting the current method of removing the VxDMP layer altogether is not flexible. Because, this takes away the multipathing functionality provided by VxDMP for other devices in the system.

Additionally two new features (Co-existence with additional third party multipathing solutions and Platform Independent Device Naming) to be added to the next release of VxVM require that the vxdmp driver should always be present in the system.

This write up outlined a new method, which will give the user ability to do the following without removing the VxDMP layer:

- **Mask certain or all devices from being autoconfigured by VxVM.**
- **Prevent VxVM (VxDMP) from multipathing certain or all devices on the system.**

Usage model

The user was presented with interfaces through `vxinstall` and `vxdiskadm` to do the following;

- Suppress devices from VxVM's view.
- Prevent devices from being multipathed by VxDMP.
- Unsuppress devices that were previously suppressed from VxVM's view

Allow multipathing by VxDMP for devices that were earlier prevented from being multipathed by it.

The user can specify the devices for the above operations using the following mechanisms:

- Using a `VendorID:ProductID` combination, e.g. `EMC:SYMMETRIX`
- Using controller name, e.g. `c1`
- Specifying path names, e.g. `c1t0d9`
- All devices on the system

Files and Directories

```
# ll /usr/sbin/vx*
-r-xr-xr-x 1 bin bin 589824 Nov 1 21:52 /usr/sbin/vxassist
lrwxr-xr-x 1 bin bin 15 Jan 3 15:53 /usr/sbin/vxconfigd ->
/sbin/vxconfigd
lrwxr-xr-x 1 bin bin 12 Jan 3 15:53 /usr/sbin/vxdctl ->
/sbin/vxdctl
-r-xr-xr-x 1 bin bin 475136 Nov 1 21:06 /usr/sbin/vxdg
lrwxr-xr-x 1 bin bin 12 Jan 3 15:53 /usr/sbin/vxdisk ->
/sbin/vxdisk
-r-xr-xr-x 1 bin bin 3221 Oct 12 15:59 /usr/sbin/vxdiskadd
-r-xr-xr-x 1 bin bin 2135 Oct 12 15:59 /usr/sbin/vxdiskadm
lrwxr-xr-x 1 bin bin 14 Jan 3 15:53 /usr/sbin/vxdmpadm ->
/sbin/vxdmpadm
lrwxr-xr-x 1 root sys 24 Dec 9 01:34 /usr/sbin/vxdump ->
/usr/sbin/vxdump
lrwxr-xr-x 1 bin bin 12 Jan 3 15:53 /usr/sbin/vxedit ->
/sbin/vxedit
-r-xr-xr-x 1 bin bin 143360 Nov 1 21:48 /usr/sbin/vxinfo
-r-xr-xr-x 1 bin bin 4842 Oct 12 15:59 /usr/sbin/vxinstall
lrwxr-xr-x 1 bin bin 11 Jan 3 15:53 /usr/sbin/vxiiod ->
/sbin/vxiiod
-r-xr-x--x 1 bin bin 127797 Nov 1 04:57 /usr/sbin/vxliccheck
lrwxr-xr-x 1 root root 15 Dec 9 01:46 /usr/sbin/vxlicense ->
/sbin/vxlicense
lrwxr-xr-x 1 bin bin 12 Jan 3 15:53 /usr/sbin/vxmake ->
/sbin/vxmake
lrwxr-xr-x 1 bin bin 12 Jan 3 15:53 /usr/sbin/vxmend ->
/sbin/vxmend
-r-xr-xr-x 1 bin bin 126976 Nov 1 21:13 /usr/sbin/vxnotify
lrwxr-xr-x 1 bin bin 12 Jan 3 15:53 /usr/sbin/vxplex ->
/sbin/vxplex
lrwxr-xr-x 1 bin bin 13 Jan 3 15:53 /usr/sbin/vxprint ->
/sbin/vxprint
lrwxr-xr-x 1 bin bin 15 Jan 3 15:53 /usr/sbin/vxrecover ->
/sbin/vxrecover
lrwxr-xr-x 1 bin bin 16 Jan 3 15:53 /usr/sbin/vxrelayout ->
/sbin/vxrelayout
lrwxr-xr-x 1 root sys 27 Dec 9 01:34 /usr/sbin/vxrestore ->
/usr/sbin/vxrestore
lrwxr-xr-x 1 bin bin 10 Jan 3 15:53 /usr/sbin/vxsd ->
/sbin/vxsd
-r-xr-xr-x 1 bin bin 106496 Nov 1 21:25 /usr/sbin/vxstat
-r-xr-xr-x 1 bin bin 102400 Nov 1 21:27 /usr/sbin/vxtask
-r-xr-xr-x 1 bin bin 139264 Nov 1 21:20 /usr/sbin/vxtrace
-r-xr-xr-x 1 bin bin 2418 Oct 12 15:59 /usr/sbin/vxvmconvert
lrwxr-xr-x 1 bin bin 11 Jan 3 15:53 /usr/sbin/vxvol ->
/sbin/vxvol

# ll /etc/vx/bin/*
lrwxr-xr-x 1 bin bin 13 Jan 3 15:53 /etc/vx/bin/egettxt ->
/sbin/egettxt
-r-xr-xr-x 1 bin bin 28672 Nov 1 21:21 /etc/vx/bin/strtovoff
-r-xr-xr-x 1 bin bin 16384 Nov 1 21:58 /etc/vx/bin/vxbadcxcl
-r-xr-xr-x 1 bin bin 16384 Nov 1 21:58 /etc/vx/bin/vxbaddxcl
-r-xr-xr-x 1 bin bin 16384 Nov 1 21:21 /etc/vx/bin/vxbbdir
-r-xr-xr-x 1 bin bin 5283 Nov 1 21:58 /etc/vx/bin/vxbootsetup
-r-xr-xr-x 1 bin bin 1298 Oct 12 16:15 /etc/vx/bin/vxcap-lvmpart
-r-xr-xr-x 1 bin bin 5030 Oct 12 16:15 /etc/vx/bin/vxcap-lvmvol
-r-xr-xr-x 1 bin bin 20480 Nov 1 21:23 /etc/vx/bin/vxcheckda
-r-xr-xr-x 1 bin bin 12288 Nov 1 21:24 /etc/vx/bin/vxchk4age
-r-xr-xr-x 1 bin bin 12288 Nov 1 21:24 /etc/vx/bin/vxchk4badblks
```

```

-r-xr-xr-x 1 bin bin 2460 Oct 12 16:15 /etc/vx/bin/vxchksymlinks
-r-xr-xr-x 1 bin bin 278528 Nov 1 21:26 /etc/vx/bin/vxckdiskrm
-r-xr-xr-x 1 bin bin 94208 Nov 1 21:32 /etc/vx/bin/vxclustadm
-r-xr-xr-x 1 bin bin 98304 Nov 1 21:31 /etc/vx/bin/vxclustd
-r-xr-xr-x 1 bin bin 1365 Oct 12 16:15 /etc/vx/bin/vxcntrllist
-r-xr-xr-x 1 bin bin 3569 Oct 12 16:15 /etc/vx/bin/vxcommon
-r-xr-xr-x 1 bin bin 118784 Nov 1 21:32 /etc/vx/bin/vxcvm_agent
-r-xr-xr-x 1 bin bin 16384 Nov 1 21:58 /etc/vx/bin/vxcxcltd
-r-xr-xr-x 1 bin bin 1529 Oct 12 16:15 /etc/vx/bin/vxdevlist
-r-xr-xr-x 1 bin bin 1781 Oct 12 16:15 /etc/vx/bin/vxdiskrm
-r-xr-xr-x 1 bin bin 6902 Oct 12 16:15 /etc/vx/bin/vxdisksetup
-r-xr-xr-x 1 bin bin 5333 Oct 12 16:15 /etc/vx/bin/vxdiskunsetup
-r-xr-xr-x 1 bin bin 3125 Oct 12 16:15 /etc/vx/bin/vxdmpdis
-r-xr-xr-x 1 bin bin 3584 Oct 12 16:15 /etc/vx/bin/vxdmpen
-r-xr-xr-x 1 bin bin 16384 Nov 1 21:58 /etc/vx/bin/vxdxcltd
-r-xr-xr-x 1 bin bin 2176 Oct 12 16:15 /etc/vx/bin/vxedvtoc
-r-xr-xr-x 1 bin bin 3842 Oct 12 16:15 /etc/vx/bin/vxevvac
-r-xr-xr-x 1 bin bin 110592 Nov 1 21:57 /etc/vx/bin/vxhpcap
-r-xr-xr-x 1 bin bin 24576 Nov 1 21:57 /etc/vx/bin/vxislvm
-r-xr-xr-x 1 bin bin 16384 Oct 12 16:17 /etc/vx/bin/vxldiskcmd
-r-xr-xr-x 1 bin bin 10747 Oct 12 16:15 /etc/vx/bin/vxlvmenclap
-r-xr-xr-x 1 bin bin 2114 Oct 12 16:15 /etc/vx/bin/vxlvmgvlist
-r-xr-xr-x 1 bin bin 5496 Oct 12 16:15 /etc/vx/bin/vxmirror
-r-xr-xr-x 1 bin bin 20480 Nov 1 21:57 /etc/vx/bin/vxmktvoc
-r-xr-xr-x 1 bin bin 2581 Oct 12 16:15 /etc/vx/bin/vxnewdmname
-r-xr-xr-x 1 bin bin 20480 Nov 1 21:57 /etc/vx/bin/vxparms
-r-xr-xr-x 1 bin bin 6251 Oct 12 16:15 /etc/vx/bin/vxpfto
-r-xr-xr-x 1 bin bin 2063 Oct 12 16:15 /etc/vx/bin/vxpvtvoc
-r-xr-xr-x 1 bin bin 5263 Oct 12 16:15 /etc/vx/bin/vxr5check
-r-xr-xr-x 1 bin bin 32768 Nov 1 21:57 /etc/vx/bin/vxr5vrfy
-r-xr-xr-x 1 bin bin 20480 Nov 1 21:58 /etc/vx/bin/vxrdvtoc
-r-xr-xr-x 1 bin bin 6060 Oct 12 16:15 /etc/vx/bin/vxreattach
-r-xr-xr-x 1 bin bin 44917 Oct 12 16:15 /etc/vx/bin/vxrelocd
-r-xr-xr-x 1 bin bin 45056 Nov 1 21:57 /etc/vx/bin/vxresize
-r-xr-xr-x 1 bin bin 3676 Oct 12 16:15 /etc/vx/bin/vxresync
-r-xr-xr-x 1 bin bin 2103 Oct 12 16:15 /etc/vx/bin/vxroot
-r-xr-xr-x 1 bin bin 6574 Oct 12 16:16 /etc/vx/bin/vxsetup
-r-xr-xr-x 1 bin bin 24576 Nov 1 21:58 /etc/vx/bin/vxspare
-r-xr-xr-x 1 bin bin 13342 Oct 12 16:16 /etc/vx/bin/vxsparecheck
-r-xr-xr-x 1 bin bin 3659 Oct 12 16:16 /etc/vx/bin/vxstartup
-r-xr-xr-x 1 bin bin 1999 Oct 12 16:16 /etc/vx/bin/vxtaginfo
-r-xr-xr-x 1 bin bin 11651 Oct 12 16:15 /etc/vx/bin/vxunreloc
-r-xr-xr-x 1 bin bin 17547 Oct 12 16:16 /etc/vx/bin/vxunroot
-r-xr-xr-x 1 bin bin 352256 Nov 1 21:58 /etc/vx/bin/vxvmbot
-r-xr-xr-x 1 bin bin 2931 Oct 12 16:16 /etc/vx/bin/vxwatch

```

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 namedg disk01=c1t1d0 disk02 ...
List disk groups	vxdg list ([namedg] for more info)
Add disk to group	vxdg -g namedg adddisk disk01=c1t1d0 disk02 ...
Remove disk from group	vxdg -g namedg rmdisk disk01
Deport disk group	vxdg deport namedg
Import disk group	vxdg import namedg (may then have to vxrecover -g namedg -sb)
Destroy disk group	vxdg destroy namedg
See free space	vxassist -g namedg maxsize [layout= ...]
Volume Operations	
Create a volume	vxassist -g namedg make volname size layout=format disk01 disk02 ...
Remove a volume	vxedit -g namedg [-rf] rm volname
Rename a volume	vxedit -g namedg rename volname newvolname
Resize a volume	vxassist -g namedg 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 namedg start[stop] volname vxvol -g namedg startall [stopall (be careful)] vxrecover -sn volname
Move volume to another disk	vxassist -g namedg move volname !disk01 disk02
Change layout: add another stripe column add mirror remove mirror	vxassist -g namedg relay layout volname ncol=... vxassist -g namedg mirror volname vxplex dis <plexname> 0 vxplex -o rm dis <plexname> to remove plex as well
Snapshot a volume	vxassist -g namedg snapstart volname vxassist -g namedg snapshot volname newvolname
Add log to a volume	vxassist -g namedg addlog volname [alloc=disk ..]
Remove log	vxplex -g namedg -o rm dis volname
Evacuate disk	vxevac -g namedg <fromdisk> <todisk>

Plex Operations	
Create a plex	<code>vxmake -g namedg plex plexname sd=subdiskname</code>
Associate a plex to a vol	<code>vxplex -g namedg att volname plexname</code>
Dis-associate a plex	<code>vxplex dis plexname</code>
Remove a plex	<code>vxedit -g namedg rm plexname</code>
List all plexes	<code>vxprint -lp</code>
Detach a plex	<code>vxplex -g namedg det plexname</code>
Subdisk Operations	
Create a subdisk	<code>vxmake -g namedg sd sdname disk,offset,length</code>
Remove a subdisk	<code>vxedit -g namedg rm sdname</code>
Associate a subdisk to a vol	<code>vxsd assoc plexname sdname</code>
Dis-associate a subdisk	<code>vxsd dis sdname</code>
Join subdisk	<code>vxsd join sd1 sd2 newsdname</code>
Recovery	
Save config data to file Look at dump Restore config	<code>vxprint (-g namedg) -Qqmr > file</code> [after you've rebuilt rootdg] <code>cat file vxprint -d --ht</code> <code>vxmake -g namedg -d file</code> <code>vxvol start volname</code>
Plex attach, raid5 subdisk recovery, resync	<code>vxrecover</code> or <code>vxplex att</code>
Start volume	<code>vxvol (-f) start volname</code>
Set plex states	<code>vxmend fix (clean active stale empty) plex</code>
Initialise a volume	<code>vxvol init (active clean enable) volname</code>
Re-attaching a failed disk	<code>vxdctl enable</code> <code>vxdg -g namedg -k adddisk faileddiskname=cntndn</code> <code>vxrecover -g namedg -s faileddiskname</code> <code>/etc/vx/bin/vxbootsetup faileddiskname</code>
Non-VM boot disk failure where rootdg is on other disk(s)	Replace boot disk Install OS Install VxVM (don't do <code>vxinstall</code>) Add license keys <code>rm /etc/vx/reconfig.d/state.d/install-db</code> <code>vxiod set 10</code> <code>vxconfig -d</code> (ie. run in disabled mode) <code>vxdctl init hostname</code> (creates <code>/etc/vx/volboot</code>) <code>vxdctl enable</code> 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 <code>vxinstall</code> Reboot
Dynamic Multipathing (DMP)	
Show all controllers	<code>vxdmadm listctlr all</code>
Enable/disable controller	<code>vxdmadm enable[disable] ctlr=.....</code>
Stop & restart <code>vxconfigd</code>	<code>vxconfigd -k</code>
Disable/enable <code>vxconfigd</code>	<code>vxconfigd -k -m disable[enable]</code>

Turn on logging [to a file]	<code>vxconfigd -k [-x logfile=/tmp/log] -x 1 (up to 9)</code>
To redo vxinstall	<code>touch /etc/vx/reconfig.d/state.d/install-db</code>

Additional information

VERITAS Volume Manager 3.1 Administrator's Guide 11.i

<http://www.docs.hp.com/hpux/onlinedocs/B7961-90002/B7961-90002.html>

VERITAS Volume Manager 3.1 Migration Guide:

<http://www.docs.hp.com/hpux/onlinedocs/B7961-90017/B7961-90017.html>

WTEC JFS/VxFS Web Site:

<http://wtec.cup.hp.com/~hpux/fs/vxvm.htm> (HP internal)

VxVM-Training Module 0–10:

http://hpuxse.cup.hp.com/Training/11i/200/Veritas/VxVM_3.1 (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/dmp_and_emc_power_path_exclusive.htm (HP internal)

New Features of VxVM 3.11 for DMP:

<http://seer.support.veritas.com/docs/235981.htm> (non HP)

VxVM Internals:

http://hpuxse.cup.hp.com/Training/11i/400/Veritas/VxVM_3.1/Vm_internals.pdf (HP internal)

PDF document overview:

http://ftp.support.veritas.com/pub/support/products/VolumeManager_UNIX/ (non HP)

Integrating VERITAS Volume Manager (VxVM) with MC/ServiceGuard A.11.09:

<http://docs.hp.com/hpux/onlinedocs/B3936-90048/B3936-90048.html>