

Veritas Volume Manager Conversion

Tim Clemens



Training and Consulting

Louisville, KY

1-800-888-9028



Topics

- Veritas Volume Manager - VxVM
- LVM Cluster Volume Groups - Review
- Creating VxVM Cluster Disk Groups
- Package Configuration Changes
- Converting Volume Groups to Disk Groups

HP-UX 11i Versions

- HP-UX 11i 1.2 (B.11.11)
 - PA-RISC
 - MC/SG 11.14
 - VxVM 3.1
 - CVM 3.2
- HP-UX 11i 1.5 (B.11.20)
 - Itanium Only
 - No MC/SG
 - Boots VxVM
 - No boot from LVM
- HP-UX 11i 1.6 (B.11.22) (6/10/02)
 - Itanium Only
 - MC/SG

Volume Manager Choices

		Base	Addon	
	LVM	VxVM	VxVM	CVM
16 Node Cluster	Yes	Yes	Yes	4 Nodes
Alternate-Links	Standby	No	Active	Active
Exclusive Activation	Yes	Yes	Yes	Yes
Shared Activation	Yes	No	No	Yes+
Cluster lock disk	Yes	No	No	No
Multiple Heartbeat Subnets	Yes	Yes	Yes	No
Striped Mirrors or RAID5	No	No	Yes	Yes
Additional Licenses	No	No	Yes	Yes

Primary Benefits of VxVM

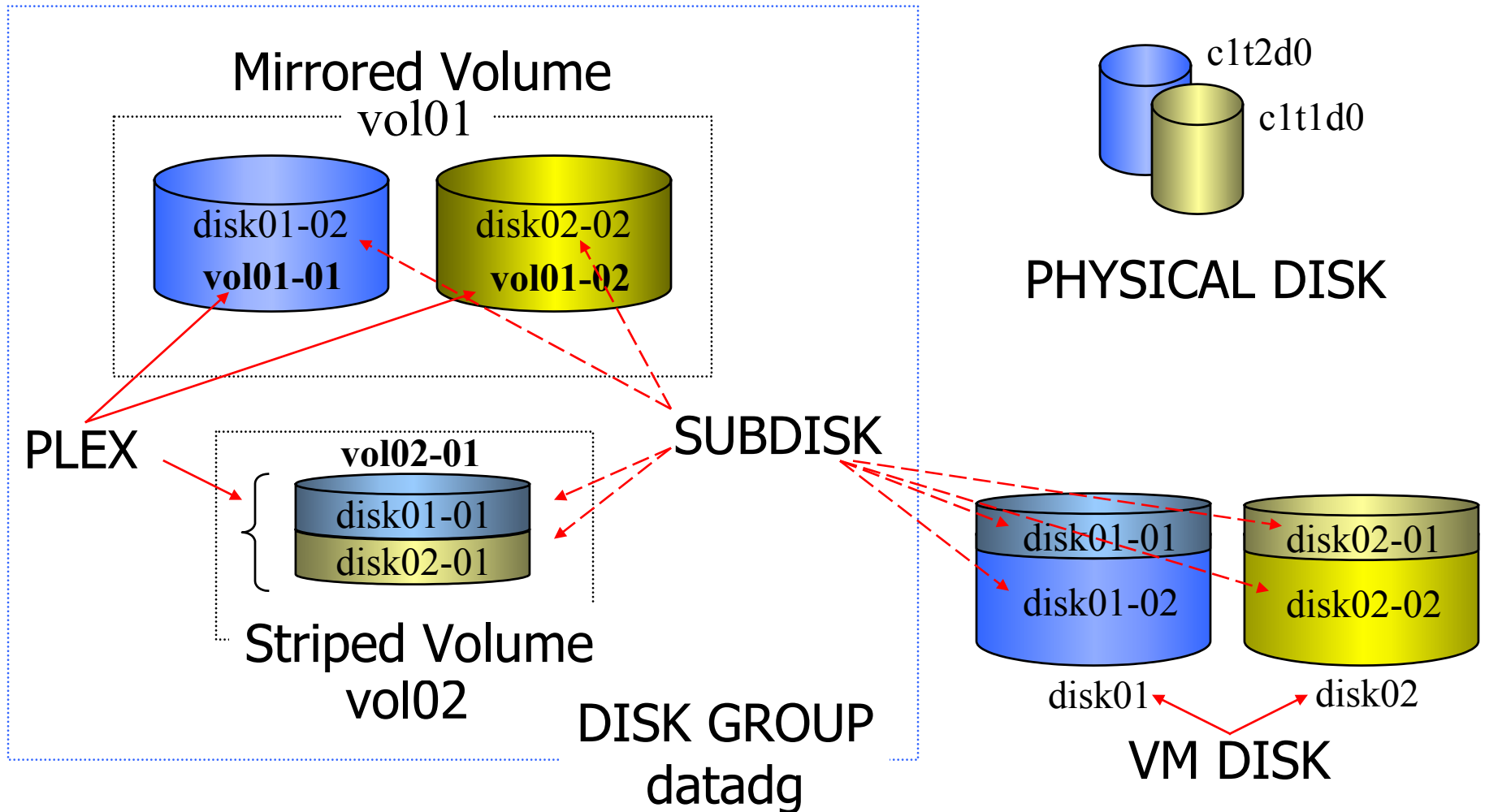
B9116AA - Additional License

- Dynamic Multipathing (DMP)
 - Active/Active primary and alternate paths (load balancing)
- Online Reallocation
- Hot Relocation
- Fast Mirror Resynchronization (FMR)
- Online Backup

Drawbacks of VxVM with SG

- Cannot be used as Cluster Lock
 - Use Quorum Server
- Must have a root disk group on every node
 - Can be use for data storage but not as a cluster volume
 - This is NOT the boot volume
- root disk group must be mirrored
 - Eliminate single point of failure

Veritas Volume Manager: Overview



Comparison of Terms

LVM

VxVM

Physical Volume

VM Disk

Volume Group

Disk Group

Logical Volume

Volume or Plex

- A VxVM Subdisk is a contiguous area of a VM Disk. Like an extent but much larger and can be varying sizes.
- A VxVM Plex is like a Logical Volume. It includes all mirrored/striped areas of various disks.
- LVM header => Private Region
- Export => Deport

Install and Initialize VxVM

All nodes of the cluster

- Install VxVM using swinstall
- Initialize VxVM with vxinstall utility
 - initializes directory structures
 - select disks for the root disk group
 - searches all I/O channels for all devices
 - prompts user after each I/O channel scan
 - detects disks with LVM headers
 - do NOT use the Quick Install...use custom

LVM Cluster Volume Groups - Review

- Create VG on any node of the cluster
 - `mkdir /dev/vg01_A; mknod /dev/vg01_A/group c 64 0x1a0000`
 - `pvcreeate /dev/rdisk/c2t12d0` (repeat for each disk)
 - `vgcreate vg01_A /dev/dsk/c2t12d0 /dev/dsk/c1t12d0 ...`
- Create needed Logical Volumes and Filesystems
 - `lvcreate -L 1024 -m 1 -n A_lvol1 vg01_A`
 - `newfs -F vxfs /dev/vg01_A/rA_lvol1`
- Deactivate and set cluster flag (cluster must be running)
 - `vgchange -a n vg01_A; vgchange -c y vg01_A`
- Pretend to export VG creating Map file
 - `vgexport -v -p -s -m /tmp/vg01_A.map vg01_A`

LVM Cluster Volume Groups - Review

Continued

- Import VG onto other nodes that may run the package (package nodes)
 - `mkdir /dev/vg01_A; mknod /dev/vg01_A/group c 64 0x1a0000`
 - `vgimport -v -s -m /tmp/vg01_A.map vg01_A`
- Create needed mount points on package nodes
 - `mkdir /mntpnt`

Volume Groups are already “known” to all package nodes. Their Name, VGID, and disks are listed in `/etc/lvmtab`. Their devices files are created in `/dev/<vgname>`. Package control scripts “activate” them with `vgchange`.

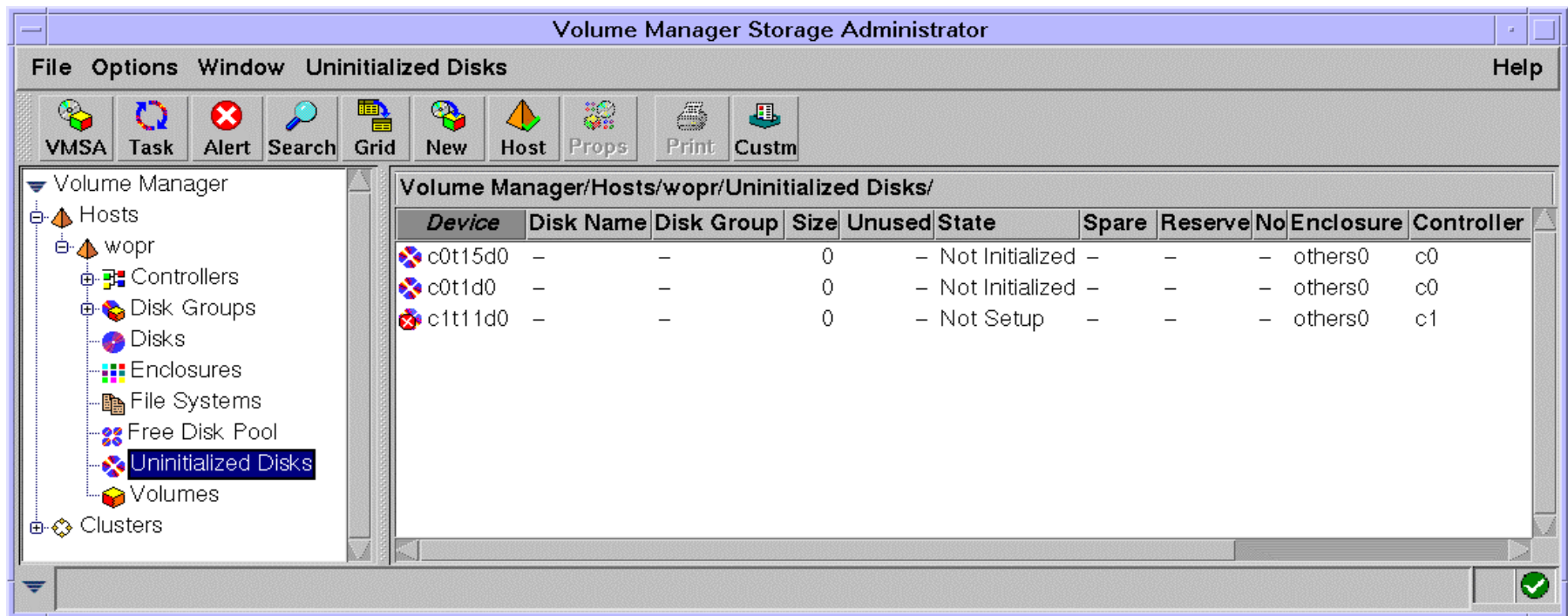
Creating VxVM Cluster Disk Groups

- Initialize Disks
 - `vxdisksetup -i c0t9d0` (repeat for each) (`/usr/lib/vxvm/bin`)
 - Use `pvremove` to remove the LVM header
 - May need `pvcreate -f` and then `pvremove`
- Create Disk Group
 - `vx dg init dg01_A c0t9d0 c1t9d0`
 - To view use: `vx dg list` and/or `vxprint -g <diskgroup>`
- Create Volumes and Filesystems
 - `vxassist -g dg01_A make A_vol01 2048m`
 - `newfs -F vxfs /dev/vx/rdisk/dg01_A/A_vol01`

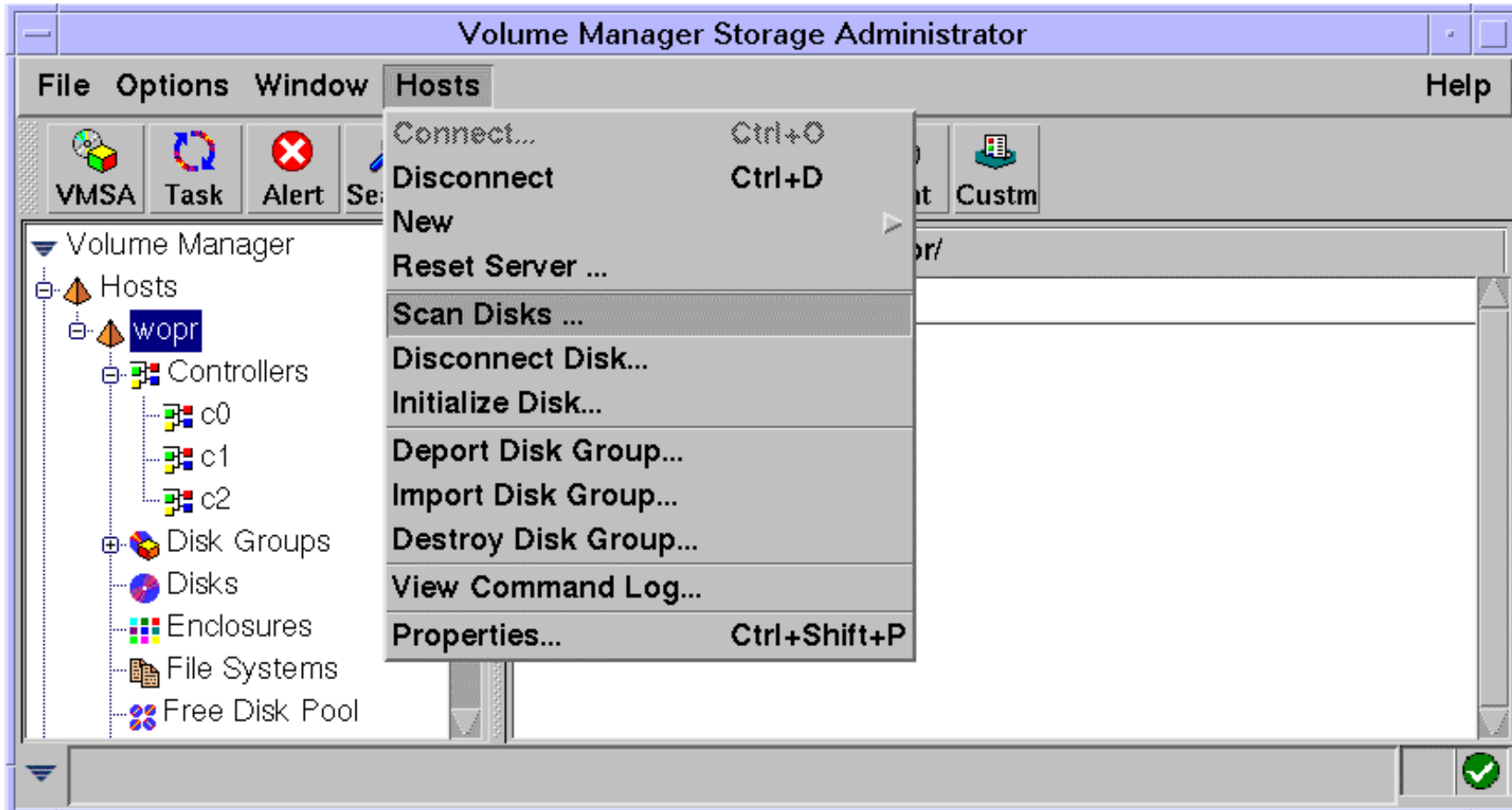
Veritas Volume Manager Storage Administrator - VMSA

- Start the VMSA server
 - `nohup /opt/HPvmsa/bin/vmsa_server &`
- Stop the VMSA server
 - `/opt/HPvmsa/bin/vmsa_server -k`
- To configure auto start of VMSA server at boot-time
 - `/opt/HPvmsa/bin/autostart on`
- Java VMSA GUI
 - `/opt/HPvmsa/bin/vmsa`

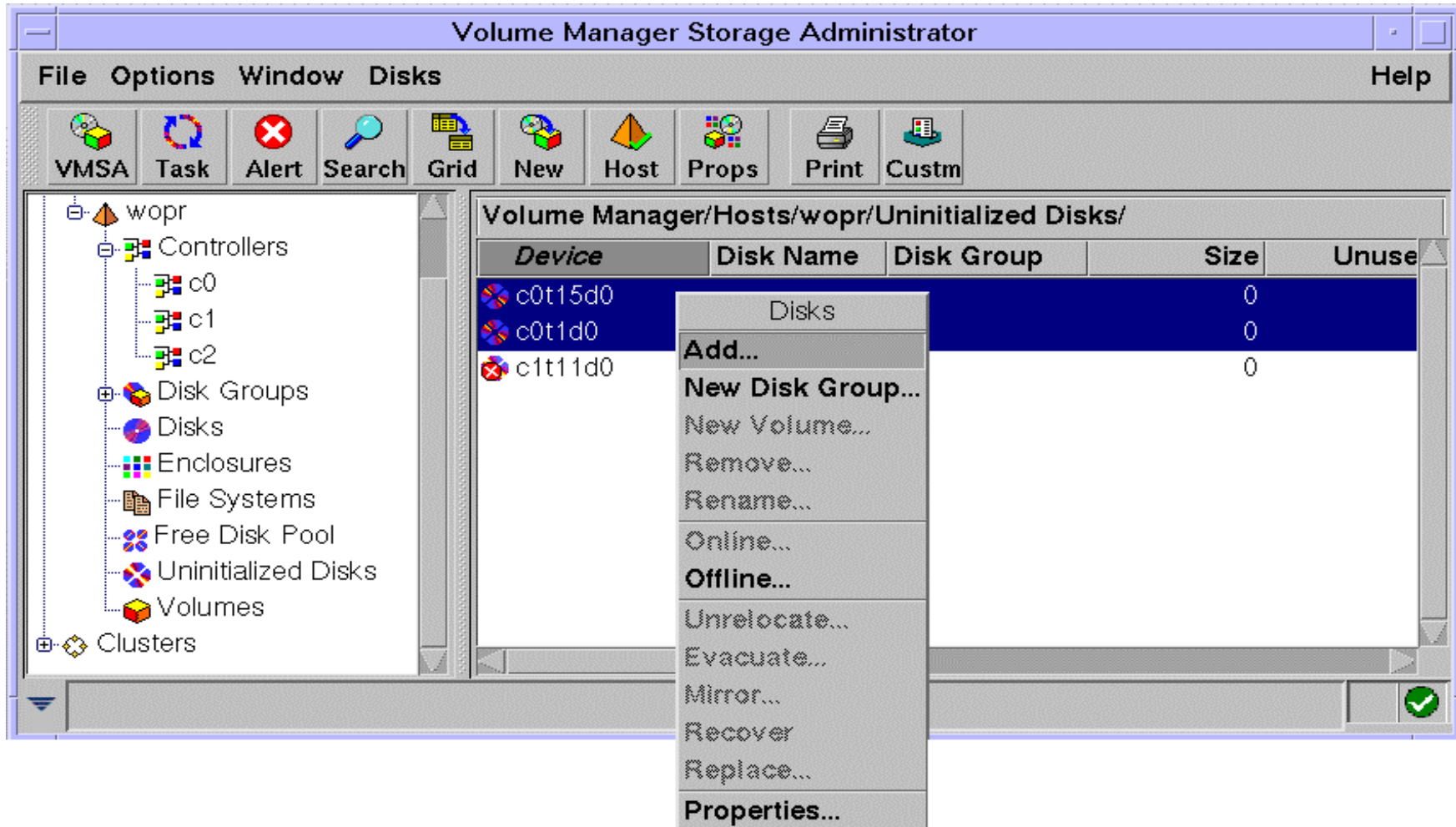
Select Uninitialized Disks



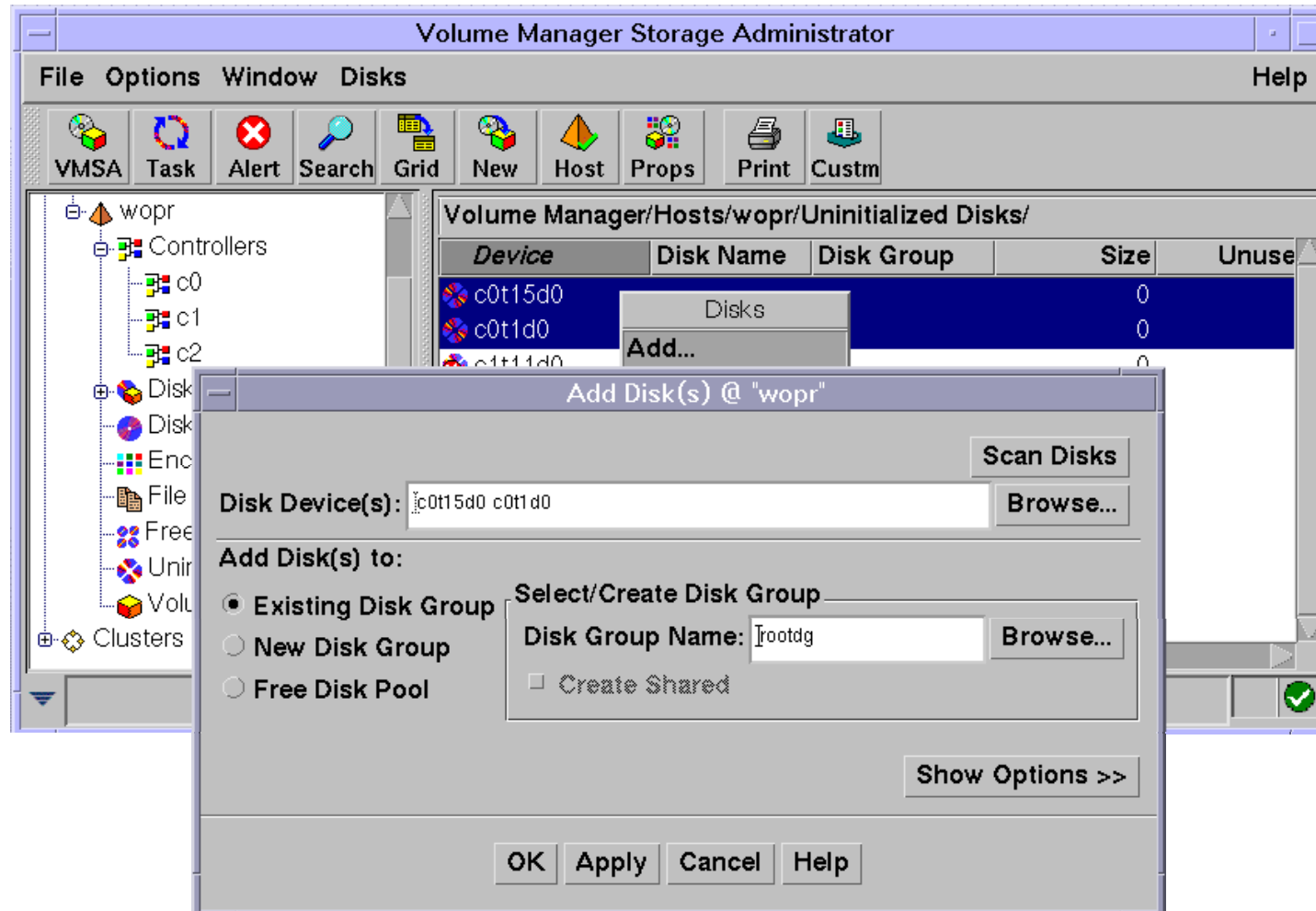
Scan for Added Disks



Select: Disks >> Add



Specify Disk Group



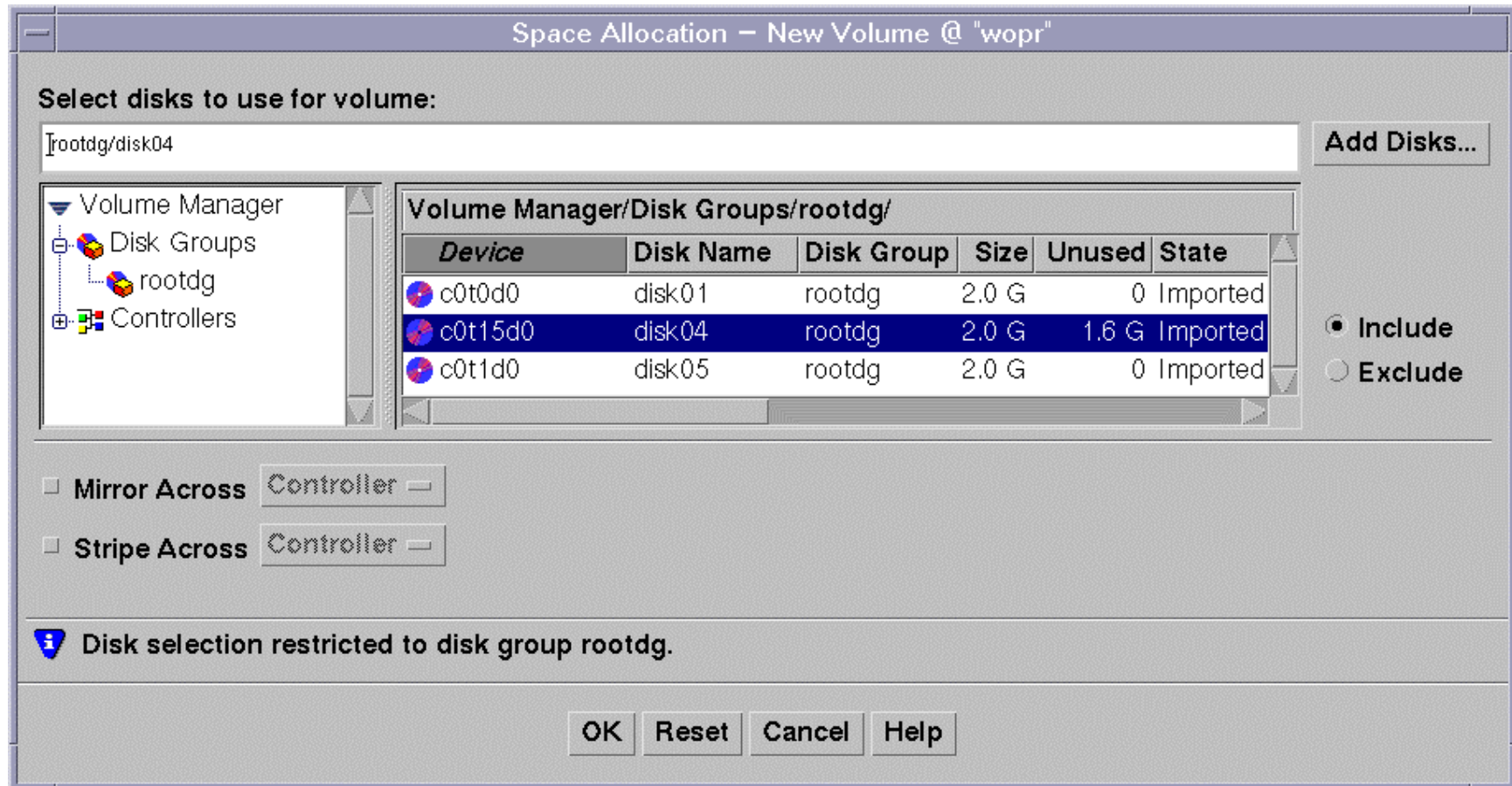
Create a Volume

File >> New >> Volume

The screenshot shows a 'New Volume' dialog box with the following fields and options:

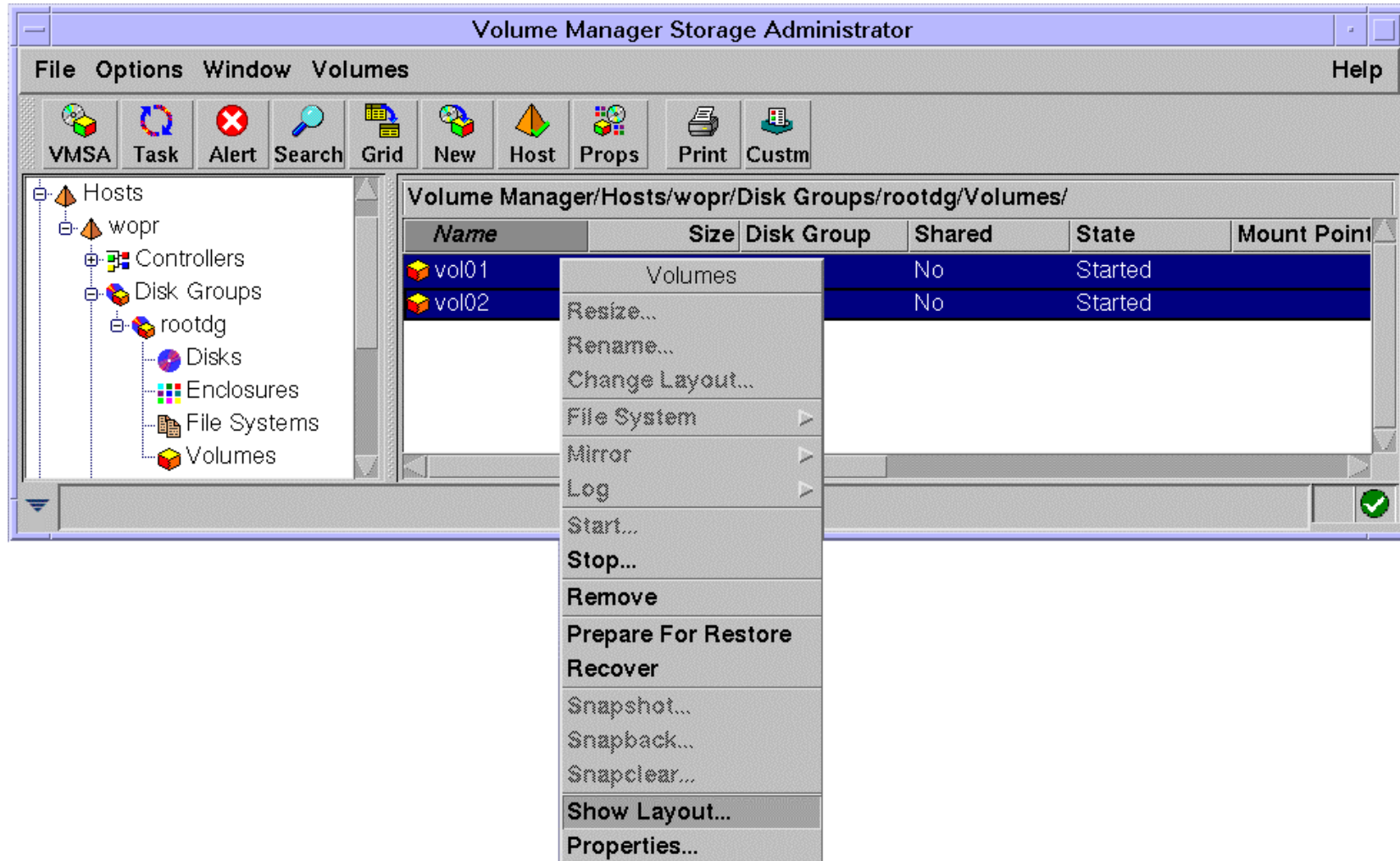
- Disk Group Name:** rootdg (with a 'Browse...' button)
- Volume Name:** vol01
- Comment:** Project Data Volume
- Size:** 4500M (with a 'Maxsize' button)
- Layout:**
 - ☒ Concatenated
 - ☐ Striped (with 'Number of Columns: 1' and 'Stripe Unit Size: 1M')
 - ☐ RAID-5
 - ☐ Concatenated Pro
 - ☐ Striped Pro
- Mirror Info:**
 - ☐ Mirrored (with 'Total Number of Mirrors: 1')
 - ☐ Enable Logging
 - ☐ Initialize Zero
 - ☐ No Layered Volumes
- Buttons:** 'Assign Disks...', 'Add File System...' (with 'File System: None' dropdown), 'OK', 'Apply', 'Reset', 'Cancel', and 'Help'.

Can Assign Volume to Specific Disks



View Volume Layout

Select: Volumes >> Show Layout...



Volume Layout View

The screenshot displays two windows from a storage management application. The top window, titled 'Volume 'vol01' Layout Details @ 'wopr'', shows a 'Concat Plex' configuration for 'vol01' (3.0 G). It lists two subdisks: 'disk01-01' on 'disk01' (2.0 G, offset 0) and 'disk02-01' on 'disk02' (1.0 G, offset 2.0 G). The bottom window, titled 'Volume 'vol02' Layout Details @ 'wopr'', shows a 'Striped Plex' configuration for 'vol02' (1.0 G, 2x64 Col/Width). It lists two subdisks: 'disk03-01' on 'disk03' (512.0 M, offset 0, column 0) and 'disk02-02' on 'disk02' (512.0 M, offset 0, column 1).

Volume 'vol01' Layout Details @ 'wopr'

File View Show/Hide Help

Volume: vol01, Size: 3.0 G
Concat Plex: vol01-01, Size: 3.0 G

Subdisk	Disk	Plex	Length	Offset	Column	Start
disk01-01	disk01	vol01-01	2.0 G	0	0	0
disk02-01	disk02	vol01-01	1.0 G	2.0 G	0	0

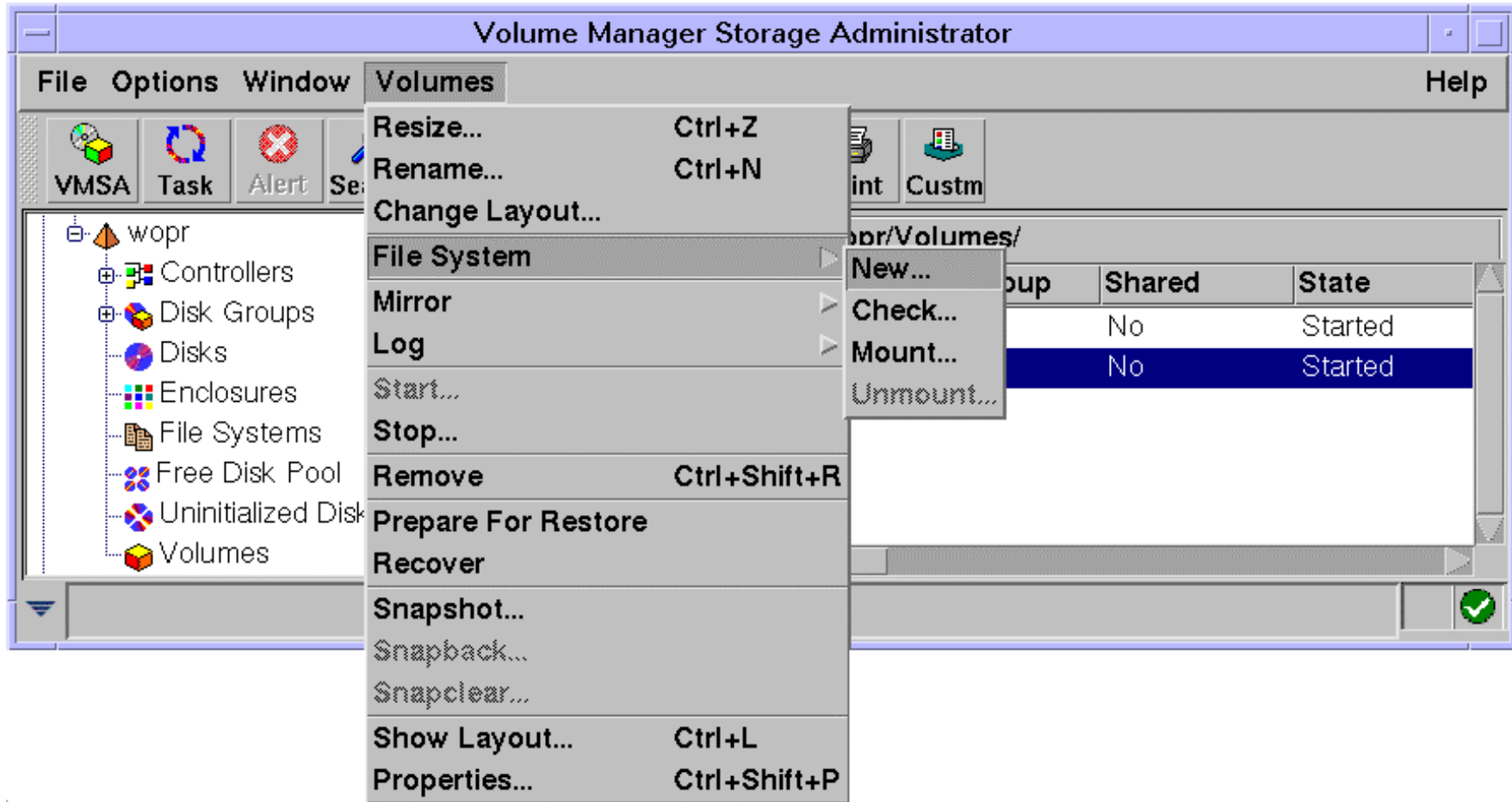
Volume 'vol02' Layout Details @ 'wopr'

File View Show/Hide Help

Volume: vol02, Size: 1.0 G
Striped Plex: vol02-01, Size: 1.0 G, Col/Width: 2x64

Subdisk	Disk	Plex	Length	Offset	Column	Start
disk03-01	disk03	vol02-01	512.0 M	0	0	0
disk02-02	disk02	vol02-01	512.0 M	0	1	1.0 G

Add a Filesystems



Add a Filesystems



Creating VxVM Cluster Disk Groups

Continued

- Deport disk group from originating node
 - `vx dg deport dg01_A`
- Enable vxconfigd on all other package nodes
 - `vxctl enable`
 - Causes vxconfigd to recognize the new disk groups
- Import new disks groups on needed nodes and test
 - `vx dg import dg01_A` – `vxvol -g dg01_A startall`
 - `mkdir /mntpnt` – `mount /dev/vx/dsk/dg01_A/A_lvol1`
- After testing, deport disk groups
 - `umount /mntpnt`
 - `vx dg deport dg01_A`

Testing Volumes without the Cluster

- LVM

- `vgchange -a e <VG>` (cluster running)
- `vgchange -c n <VG>; vgchange -a y <VG>` (cluster down)
- `fsck, mount /dev/vg01_A/A_lvol1 /mntpnt`
- `umount /mntpnt`
- `vgchange -a n <VG>`
- `vgchange -c y <VG>` (when cluster is running)

- VxVM

- `vxvg import dg01_A`
- `vxvol -g dg01_A startall`
- `fsck, mount /dev/vx/dsk/dg01_A/A_vol01 /mntpnt`
- `umount /mntpnt`
- `vxvg deport dg01_A`

Package Configuration

- Only change is in the control script
 - No need to re-apply configuration
 - VXVM_DG[x]
 - VXVM_DG[0]=dg01_A
 - VXVM_DG[1]=dg01_B
 - VXVOL
 - VXVOL="vxvol -g \\${DiskGroup} [-o bg] startall"
 - LV[x], FS[x], FS_MOUNT_OPT[x]
 - FS_MOUNT_RETRY_COUNT
 - FS_UMOUNT_COUNT

Example Control Script Entries

```
VGCHANGE="vgchange -a e"
VXVOL="vxvol -g \${DiskGroup} -o bg startall"

VG[0]="vg01_A"
VG[1]="vg02_A"
VXVM_DG[0]=dg01_A
VXVM_DG[1]=dg02_A

LV[0]=/dev/vg01_A/dbA1; FS[0]=/dbA1; FS_MOUNT_OPT[0]="-o delaylog"
LV[1]=/dev/vg01_A/dbA2; FS[1]=/dbA2; FS_MOUNT_OPT[1]="-o delaylog"
LV[2]=/dev/vg02_A/dbA3; FS[2]=/dbA3; FS_MOUNT_OPT[2]="-o delaylog"
LV[3]=/dev/vg02_A/dbA4; FS[3]=/dbA4; FS_MOUNT_OPT[3]="-o delaylog"

LV[4]=/dev/vx/dsk/dg01_A/dbA5
FS[4]=/dbA5; FS_MOUNT_OPT[4]="-o delaylog"
LV[5]=/dev/vx/dsk/db01_A/dbA6
FS[5]=/dbA6; FS_MOUNT_OPT[5]="-o delaylog"
LV[6]=/dev/vx/dsk/dg02_A/dbA7
FS[6]=/dbA7; FS_MOUNT_OPT[6]="-o delaylog"
LV[7]=/dev/vx/dsk/dg02_A/dbA8
FS[7]=/dbA8; FS_MOUNT_OPT[7]="-o delaylog"
```

Limits of vxvmconvert

- Remove cluster flag from VG
`insufficient Private Space for conversion error`
- LVM system volume group (vg00) cannot be converted (vxvmconvert detects LIF as created by mkboot)
- VG containing /usr cannot be converted
- Dump & primary swap VGs cannot be converted
- Cluster lock VG cannot be converted
- Disks with bad blocks relocated in LVM's header (no bad block relocation feature in VxVM)

Issues

- VxVM private region must fit into existing LVM header regions
- Mirrored LVM volumes with MWC will be converted to VxVM with Dirty Region Logging. DRL reserves space from the user area.
- LVM header data is stored in the root filesystem
- If VG contains swap space, reboot is required to release it

Process

- Shutdown the cluster package (cmhaltpkg)
- Remove the cluster flag (vgchange -c n <vg>)
- Activate the VG (vgchange -a y <vg>)
- User data backup
- vgcfgbackup (vgcfgbackup <vg>)
- vxvmconvert (analysis...option 1)
- vxvmconvert (convert...option 2)
- deport disk group (vxdg deport <dg>)
- import / deport on remaining cluster nodes
(vxdg import <dg>; vxdg deport <dg>)

Process Continued

- Modify Package Control Script
 - Was: VG[n]="LVM Volume Group"
 - VG[0]=vgali1
 - Becomes: VXVM_DG[n]="VxVM Disk Group"
 - VXVM_DG[0]=dgali1
 - Was: LV[n]="LVM Logical Volume"
 - LV[0]=/dev/vgali1/db1_al
 - Becomes: LV[n]="VxVM Volume"
 - LV[0]=/dev/dgali1/db1_al
- Export VG definitions from remaining nodes
 - vgexport <vgname>

Rollback and Soft Links

- Can rollback if VxVM names remain unchanged
- Use name translation stored in
`/etc/vx/reconfig.d/<vgname>/<vgname>.trans`

```
#  grep -v '^#' thisfile | while read lvm vxvm; do
#      mkdir -p `dirname $lvm` > /dev/null
#      ln -s $vxvm $lvm
#  done
#
/dev/vgali1/ali1dbs /dev/vx/dsk/dgali1/ali1dbs
/dev/vgali1/rali1dbs /dev/vx/rdisk/dgali1/ali1dbs
/dev/vgali1/ali1trace /dev/vx/dsk/dgali1/ali1trace
/dev/vgali1/rali1trace /dev/vx/rdisk/dgali1/ali1trace
/dev/vgali1/db1_al /dev/vx/dsk/dgali1/db1_al
/dev/vgali1/rdb1_al /dev/vx/rdisk/dgali1/db1_al
```

Rollback

- Can rollback if VxVM names remain unchanged
- Use name translation stored in
`/etc/vx/reconfig.d/<vgname>/<vgname>.trans`

```
#  grep -v '^#' thisfile | while read lvm vxvm; do
#      mkdir -p `dirname $lvm` > /dev/null
#      ln -s $vxvm $lvm
#  done
#
/dev/vgali1/ali1dbs /dev/vx/dsk/dgali1/ali1dbs
/dev/vgali1/rali1dbs /dev/vx/rdisk/dgali1/ali1dbs
/dev/vgali1/ali1trace /dev/vx/dsk/dgali1/ali1trace
/dev/vgali1/rali1trace /dev/vx/rdisk/dgali1/ali1trace
/dev/vgali1/db1_a1 /dev/vx/dsk/dgali1/db1_a1
/dev/vgali1/rdb1_a1 /dev/vx/rdisk/dgali1/db1_a1
```

Oracle Raw Logical Volumes

- Can rollback if VxVM names remain unchanged
- Use name translation stored in
`/etc/vx/reconfig.d/<vgname>/<vgname>.trans`

```
SQL> select file_name from dba_data_files where
       file_name like '/dev/vgali1/%';
FILE_NAME
-----
/dev/vgali1/rdb1_a1
/dev/vgali1/rdb2_a1
/dev/vgali1/rdb3_a1
SQL> alter database rename file '/dev/vgali1/rdb1_a1' to
       '/dev/vx/rdisk/dgali1/db1_a1';
```