

# VxVM Administrators Reference

## By Solarum dot com

This document is a reference for Veritas Volume Manager, now known as Veritas Storage Foundation. The first part is a quick walk through of the commands used to perform a simple typical setup on a set of disks. This is just as an example of what would be done, in context, taken from a real world setup. The next part is the full reference. This is a breakdown of all of the VxVM commands, what they do and the applicable syntax. They are grouped by the different types of Operations like Disk, Volume, Plex and so forth. I have tried to include different command formats when I found them, as well as any other information that was available. Please let me know if you find something that seems incorrect or missing by going to the Solarum contact page at <http://www.solarum.com/contact/>.

If you want to play with Veritas Storage Foundation, Symantec (they bought Veritas) now has a free version available for download for testing, development and educational purposes. Go get your copy at:  
<http://www.symantec.com/business/theme.jsp?themeid=sfbasic>

### Quick Setup Walkthrough

Here we have a list of commands that one might use to setup VxVM on their server. We will work with the following assumptions: We are setting four disks in a RAID5 array and they are locally connected SCSI.

```
# vxdisksetup -i c0t0d0 // Initialize the disk
# vxdisksetup -i c0t1d0
# vxdisksetup -i c0t2d0
# vxdisksetup -i c0t3d0
# vxdg init sysdg c0t0d0s2 c0t1d0s2 c0t2d0s2 c0t3d0s2 // Create the diskgroup
# vxassist -g sysdg make sysvol 50g layout=raid5 // Create a volume in the diskgroup
# mkfs -F vxfs -o largefiles /dev/vx/rdisk/sysdg/sysvol // Format or create the filesystem
```

\* Note: If you are using Sun Cluster, you need to add the device group to the cluster before you can format it. If you are adding a disk to an existing device group, you must refresh the devices in the cluster device group before you can format them.

If you were adding a new device group, it would look like this:

```
# scconf -a -D type=vxvm,name=sysdg,nodelist=server1:server2,failback=disabled
```

If you were simply syncing an existing group, it would look like this:

```
# scconf -c -D name=sysdg,sync
```

You could then issue the mkfs command used above to create a filesystem on your volumes.

### Disk Operations

Function	Command
Evacuate a disk	<code>vxevac -g diskgroup from _disk to _disk</code>
Initialize a disk	<code>vxdisksetup -i device</code> <code>vxdiskadd device</code> <code>vxdiskadm</code>
List all disks (local and remote)	<code>vxdisk -o alldgs list</code>
List a disk header	<code>vxdisk list diskname   device</code>
Mirror disks	<code>vxmirror -g diskgroup diskname diskname_mirror</code>
Mirror root disk	<code>vxbootsetup</code>
Rename a disk (must be part of a diskgroup)	<code>vxedit -g diskgroup rename oldname newname</code>
Rescan for disks	<code>vxdctl enable</code>

Set spare, no hot relocation, or reserved space on a disk.	<code>vxedit -g <i>diskgroup</i> set {spare nohotuse reserve}=on off <i>diskname</i></code>
Show disk statistics	<code>vxstat -g <i>diskgroup</i> -d [ <i>device_name</i> ]</code>
Show RAID5 statistics	<code>vxstat -g <i>diskgroup</i> -f MWF [ <i>device_name</i> ]</code>
Trace disk operations	<code>vxtrace -o disk [ <i>device_name</i> ]</code>
Unrelocate a disk	<code>vxunreloc -g <i>diskgroup</i> <i>original_diskname</i></code>
Un-initialize a disk	<code>vxdiskunsetup <i>device</i></code> <code>vxdisk rm <i>device</i></code>
Un-initialize a disk forcefully	<code>vxdiskunsetup -C <i>device</i></code>
<b>Disk Group Operations</b>	
<b>Function</b>	<b>Command</b>
Add a disk to disk group	<code>vx dg -g <i>diskgroup</i> adddisk <i>diskname</i>=<i>device</i></code>
Create a disk group	<code>vx dg init <i>diskgroup</i> <i>diskname</i>=<i>device</i></code>
Deport, import or destroy a disk group	<code>vx dg {deport   import   destroy } <i>diskgroup</i></code>
Join disk groups	<code>vx dg join <i>sourcedg</i> <i>targetdg</i></code>
List disk groups	<code>vx dg list [<i>diskgroup</i>]</code>
List objects affected by a disk group move operation	<code>vx dg listmove <i>sourcedg</i> <i>targetdg</i> <i>object</i></code>
Move and object between disk groups	<code>vx dg move <i>sourcedg</i> <i>targetdg</i> <i>object</i></code>
Remove a disk from a disk group	<code>vx dg -g <i>diskgroup</i> rmdisk <i>diskname</i></code>
Rewrite disk headers, configuration copies, and kernel log copies in a disk group	<code>vx dg flush <i>diskgroup</i> <i>targetdg</i> <i>object</i></code> <code>vx dctl enable (<i>for all disk groups</i>)</code>
Show free/spare space pool	<code>vx dg -g <i>diskgroup</i> {free   spare}</code>
Split objects between disk groups	<code>vx dg split <i>sourcedg</i> <i>targetdg</i> <i>object</i></code>
Upgrade disk group version	<code>vx dg [-T <i>version</i> ] upgrade <i>diskgroup</i></code>
<b>Volume Operations</b>	
<b>Function</b>	<b>Command</b>
Abort a snapshot	<code>vx assist -g <i>diskgroup</i> snapabort <i>orig_vol_name</i></code>
Add a log to a volume	<code>vx assist -g <i>diskgroup</i> addlog <i>vol_name</i></code>
Change a volume read policy	<code>vx vol -g <i>diskgroup</i> rdpol round <i>vol_name</i></code> <code>vx vol -g <i>diskgroup</i> rdpol prefer <i>vol_name</i> <i>preferred_plex_name</i></code> <code>vx vol -g <i>diskgroup</i> rdpol select <i>vol_name</i></code>
Convert to or from a layered layout	<code>vx assist -g <i>diskgroup</i> convert <i>vol_name</i> layout=<i>new_layout</i> [<i>attributes...</i>]</code>
Create a snapshot volume	<code>vx assist -g <i>diskgroup</i> -b snapstart <i>vol_name</i></code> <code>vx assist -g <i>diskgroup</i> snapshot <i>vol_name</i> <i>new_volume</i></code>
Create a volume	<code>vx assist -g <i>diskgroup</i> make <i>vol_name</i> <i>size</i> layout=<i>format</i> <i>diskname</i></code> <code>vx assist -g <i>diskgroup</i> -U fsgen make <i>vol_name</i> <i>size</i> layout=<i>format</i> <i>diskname</i></code>
Disassociate a snapshot	<code>vx assist -g <i>diskgroup</i> snapclear <i>snapshot_vol</i></code>
Display a volume	<code>vx edit -g <i>diskgroup</i> -vt <i>vol_name</i></code> <code>vx assist -g <i>diskgroup</i> -l <i>vol_name</i></code>

List un-startable volumes	<code>vxinfo [ vol_name ]</code>
Maxresize - show largest possible volume size	<code>vxassist -g diskgroup maxgrow vol_name</code>
Mirror an existing plex	<code>vxassist -g diskgroup mirror vol_name</code>
Print snapshot information	<code>vxassist -g diskgroup snapprint vol_name</code>
Reassociate a snapshot	<code>vxassist -g diskgroup snapback snapshot_vol</code>
Recover a volume	<code>vxrecover [ -g diskgroup ] -sn vol_name</code>
Relayout a volume	<code>vxassist -g diskgroup relayout vol_name layout=new_layout [attributes...]</code>
Remove a volume	<code>vxedit -g diskgroup -rf rm vol_name</code> <code>vxassist -g diskgroup remove volume vol_name</code>
Reset volume statistics	<code>vxstat -g diskgroup -r [ vol_name ]</code>
Resize a volume	<code>vxassist -g diskgroup growto vol_name new_length</code> <code>vxassist -g diskgroup growby vol_name length_change</code> <code>vxassist -g diskgroup shrinkto vol_name new_length</code> <code>vxassist -g diskgroup shrinkby vol_name length_change</code> <code>vxresize -F ufs -g diskgroup vol_name new_length</code> <code>vxvol -g diskgroup set len=value vol_name</code>
Set FastResync flag on a volume	<code>vxvol set fastresync=on vol_name</code>
Show volume statistics	<code>vxstat -g diskgroup [ vol_name ]</code>
Start/Stop volumes	<code>vxvol {start   stop} vol_name</code>
Start/Stop ALL volumes	<code>vxvol {startall   stopall}</code>
Trace volume operations	<code>vxtrace [ vol_name ]</code>
<b>Subdisk Operations</b>	
<b>Function</b>	<b>Command</b>
Associate a subdisk to a plex	<code>vxsd assoc plex_name subdisk_name</code>
Create a subdisk	<code>vxmake -g diskgroup sd subdisk_name diskname offset length</code>
Disassociate a subdisk	<code>vxsd dis subdisk_name</code>
Display subdisk information	<code>vxprint -st</code> <code>vxprint -l subdisk_name</code>
Remove a subdisk	<code>vxedit -g diskgroup rm subdisk_name</code>
<b>Plex Operations</b>	
<b>Function</b>	<b>Command</b>
Associate a plex (to a volume)	<code>vxplex -g diskgroup att vol_name plex_name</code>
Attach a plex	<code>vxplex -g diskgroup att vol_name plex_name</code>
Change state flags on a plex	<code>vxmend fix {active   clean   stale} plex_name</code>
Create a plex	<code>vxmake -g diskgroup plex plex_name sd=subdisk_name, ...</code>
Detach a plex	<code>vxplex -g diskgroup det plex_name</code>
Disassociate a plex	<code>vxplex -o rm dis plex_name</code>
Remove a plex	<code>vxedit -g diskgroup -r rm plex_name</code>
Turn plex online/offline	<code>vxmend {on   off} plex_name</code>

Benchmarking and Tuning Operations	
Function	Command
Change VxVM kernel parameters	Example: Change the VxVM kernel parameter <code>vol_max_vol</code> from the current value to a new value of 2048 by adding the parameter to the <code>/etc/system</code> file: # set vxio: vol_max_vol=2048 Then, reboot the system.
Count and size of VxVM disk I/Os completed per sample time slice to a volume	<code>vxstat -g diskgroup [ -i interval ] [-c count ] -d vol_name</code>
Sample I/O load with statistics -- random	<code>vxbench -w {rand read   rand write} -i iosize=size, iocount=count, maxfilesize=size filename</code>
Sample I/O load with statistics -- sequential	<code>vxbench -w {read   write} -i iosize=size, iocount=count, maxfilesize=size filename</code>
View currently set VxVM kernel parameters	Example: View the current setting for the kernel parameter <code>vol_max_vol</code> : # echo 'vol_max_vol/D'   adb -k
VxVM I/O trace information -- dump to and read from a file	<code>vxtrace -g diskgroup [ -t duration ] -d [ filename ] -o dev,disk vol_name; vxtrace -l -f /tmp/tracedata   pg</code>
Miscellaneous Commands	
Function	Command
Display configuration	<code>vxprint -ht</code>
License input	<code>vxlicense -c vxlicinst</code>
License read or show license	<code>vxlicense -p</code>
(Background) Task Operations	
Function	Command
Background task change speed	<code>vxtask -i task_number set slow=number</code>
Background task hold	<code>vxtask pause task_number</code>
Background task kill	<code>vxtask abort task_number</code>
Background task list	<code>vxtask -g diskgroup list</code>
Background task monitor	<code>vxtask monitor [ task_number ]</code>
Background task resume	<code>vxtask resume task_number</code>
<p><i>This reference has been brought to you by Solarum.com (<a href="http://www.solarum.com">http://www.solarum.com</a>)   Copyright © 2008, All Rights Reserved Veritas Volume Manager and Veritas Storage Foundation are Copyright © Symantec Corporation</i></p>	