**Administering Disk Group  & Disks ( Most of DG tasks can be done by  vxdiskadm )**

|  |  |  |
| --- | --- | --- |
| DG | Displaying DG Info | # vxdg list                 # vxdg list rootdg |
| DG | Displaying Free space in a DG | # vxdg free               # vxdg –g rootdg free |
| DISK | Displaying Disk Information | # Vxdisk list             #  vxdisk list  c1t0d0s2 |
| DG  DISK | Creating a Disk Group By adding a new Disk ( add disk by vxdiskadm ) | # vxdg init mktdg mkgdg01=c1t0d0s2  # vxdiskadd c1t0d0s2 ( will ask ) |
| DG  DISK | Adding a disk to a DG        ( vxdiskadm ) | # vxdiskadd c1t0d0s2 ( will ask )  or # vxdg –g Datadg adddisk datadg02= c1t0d0s2 |
| DG  DISK | Removing a dsk w/o subdisk from a DG                     ( vxdiskadm ) | # vxdg –g datadg disk02 |
| DG | Deporting a DG    ( vxdiskadm menu 9 ) | # vxvol –g datadg stopall   # vxdg deport datadg |
| DG | Importing a DG     ( vxdiskadm menu 9) | # vxdg   import <dgid>    -C will clear lock , -f will force a import  # vxrecover –g <dg\_name> -sb  # vxvol –g rootdg startall |
| DG | Temporarily renaming a DG while importing on a importing Host | # vxdg –t –n newrootdg import rootdg  -C can clear a import lock |
| DG | Permanently renaming a diskgroup while importing on a Importing Host | # vxdg  –n newrootdg import rootdg |
| DG | Renaming a DG during export | # vxdg [-h second\_host] –n new\_name deport dgname |
| DG | Temporarily removing rootdg from host1 to host2 for repair & then moving it back | 1. On host1 find the DG-ID by command : # vxdisk –s list  2. on host2 , import it with diff name : # vxdg –tC –n rootgd1 import <dgid>  3. after repair deport the Dg from host2 : # vxdg –h host1 deport rootdg1  3. Start volumes on host1   # vxrecover –g rootdg -sb |
| DG  DISK | Moving disk between DG | 1. Remove the disk from 1st DG     # vxdg rmdisk disk02  2. Add it to second DG       # vxdg –g Datadg adddisk datadg02= c1t0d0s2 |
| DG | To clear lock from a imported disk | # vxdisk clearimport disk01 |
| DG | Destroying a DG | # vxdg destroy datadg |
| DG | Joining DG (rootdg cant be source) | # vxdg join sourcedg targetdg  # vxrecover –g targetdg –sb  # vxvol –g targetdg startall |
|  | Upgradeing a DG to highest version | # vxdg upgrade data01 |
|  |  |  |
|  |  |  |
|  |  |  |

# Administering DMP

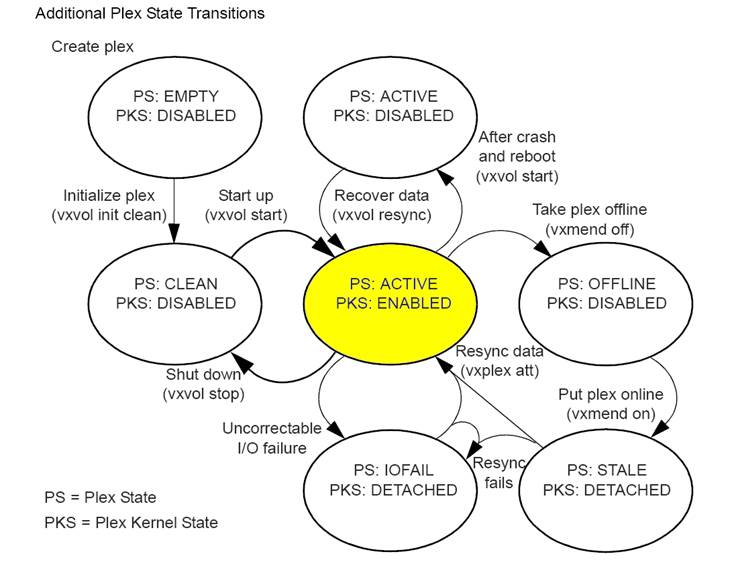
|  |  |  |
| --- | --- | --- |
| DMP  DISK | Activating Deactivating Disk,contoller enclosures | # vxdiskadm |
| DMP | Info about a DMP Node ( Disk) | # vxdmpadm getdmpnode nodename=c3t2d1s2 |
| DMP | Info about a Enclosure | # vxdmpadm getdmpnode nodename=<Encl\_name> |
| DMP | List Attributes of all Controllers | # vxdmpadm  listctlr all |
| DMP | Listing Info about Enclosures | # vxdmpadm  listenclosure all |
| DMP | Renaming a Enclosure | # Vxdmpadm setattr enclosure <current\_name> name=<new\_name> |
| DMP | Display all paths of a DMP Node | # vxdmpadm getsubpaths dmpnodename=c3t2d1s2 |
| DMP | **Display all paths through a Controller** | # vxdmpadm getsubpaths ctlr=c3 |
| DMP | Disabling a Controller | # vxdmpadm disable ctlr=c2 |
| DMP | Enabling a Controller | # vxdmpadm enable ctlr=c2 |
| DMP | Status Check of DMP Restore Daemon | # vxdmpadm stat restored |
| DMP | Status Check of DMP Error Daemon | # vxdmpadm stat errord |
| DMP | Stopping restore Daemon | # vxdmpadm stop restore   ( Same for Error Daemon ) |
| DMP | Starting The Restore daemon  Default Policy is check\_isabled  Default Interval is 300 | #vxdiskadm start restore policy=<policy\_u\_need> [interval = second]  policies are :  check\_all – Check all Paths except manually disabled. Disable down  paths & revives paths which comes  up .  check\_alternate - Checks that atleast one path is healthy ( same as “all” if only 2 paths)  check\_disabled - checks previously down paths & revive them once they come up  check\_periodic – see book ….. |

# Administering Subdisk

|  |  |  |
| --- | --- | --- |
|  | Displaying SD info | # vxprint –st         #  vxprint –l  <sd\_name> |
| SD | Creating Subdisk | Syntax is :  vxmake –g <dg\_name> sd <sd\_name> <vx\_diskname>,<**offset**>,<size>  #   vxmake –g mydg sd *disk01-01* disk01,32000,1500m  Offset can be checked by:     # vxdg –g mydg free |
| SD | Moving a SD from one disk to another | Moving sd disk03-01 to two diff SD on disk22  # vxsd mv disk03-01 disk22-01 disk22-02 |
| SD | Splitting a SD in two equal SD | Format is :  vxsd –s <size> split <sd\_name> <new\_sd1> <new\_sd2>  # vxsd –s 1000m split **disk03-01** **disk03-01** disk03-02 |
| SD | Joining subdisks | # vxsd join <sd\_1> <sd-2> <combined\_sd> |
| SD/PX | Associating SD to Plex | # vxsd –g <dgname> assoc <existing\_plex> <sd1> <sd2> <sd3> |
| SD/ PX | Creating new Concat Plex from SD | # vxmake –g <dg\_name> plex <new\_plex> sd=<sd1>,<sd2>,<sd3> |
| SD/PX | Dissociating SD from Plex | # vxsd –g <dg\_name> dis  <sd> |
| SD | Deleting a SD | # # vxsd –g <dg\_name> rm <sd> |

# Administering Plex

|  |  |  |
| --- | --- | --- |
| PX | Display Plex Info | # vxprint –lp     # vxprint –l <plex-name> |
| PX | Listing Free Plex | #vxprint -pt |
| PX | Creating new Concat Plex from SD | # vxmake –g <dg\_name> plex <new\_plex> sd=<sd1>,<sd2>,<sd3> |
| PX | Creating a Stripe Plex from SD | # vxmake  plex <new\_plex> layout=stripe stwidth=32 ncolumn=2 sd=<sd1>,<sd2> |
| VX  PX | **Attaching a Plex to an existing volume Or Re-attaching** | # vxplex  –g  <dg\_name> att <vol\_name> <plex\_name> |
| VX  PX | Creating a new volume from a Plex | # vxmake [–g <dg\_name>] -U <**usetype>** vol  <new\_vol\_name>  plex=<plex\_1>       usetype is generally :  fsgen  or raid5 |
| VX  PX | Creating a New Mirrored volume from two identical Plexes | # vxmake [–g <dg\_name>] -U **fsgen** vol  <vol\_name>  plex=<plex\_1>,.<plex\_1>,. |
| VX  PX | Making a Plex offline | # vxmend off <plex>  **# vxmend -o force off  <plex>**   To forcefully make a plex off |
| VX  PX | Detaching a Plex from a volume  ( detach but maintains the association) | # vxplex  [–g <dg\_name>] det <plex> |
| VX PX | Re-attaching a Plex to a currently  ENABLED volume | # vxplex [–g <dg\_name>]  <volume\_name> <plex>  The OFFLINE Plex will recover & will becomes ACTIVE |
| VX  PX | Putting The Plex back Online PS=STALE, PKS = **DETACHED** | # vxmend on <plex\_name>  The Plex state will be STALE, Will be ACTIVE once volume is started in case it is stopped or the  plex is resynced in case the volume is always active. |
| VX  PX | **Cleaning one of the Plex of a Unstartable volume** | # vxmend fix clean <plex\_of\_your\_choice> |
| PX | Moving content of a Plex to a New PX | # vxplex mv <original\_plex>  <new\_plex> |
| PX | Fixing and activating a STALE Plex | # vxmend fix active <plex> |
| PX | Changing the Plex state to Clean | # vxmend fix clean <plex> |
| VX  PX | Copying content of a Volume to a New Plexes | # vxplex cp <volume\_to\_be\_copied> <plex\_name>  This plex will not be attached to the volume by itself |
|  |  |  |
|  |  |  |



# Administering Volume

Creating Volume  Step By Step by administering Disk , Subdisk & Plex is already covered

Another Straight forward way is using Vxassist . By default vxassist creates a concatenated volume on available disks.

|  |  |  |
| --- | --- | --- |
| VX | Viewing Default attributes of vxassist | # vxassist help showattrs    or edit   /etc/default/vxassist |
| VX | Discover how large a volume can be created | # vxassist [-g <dg\_name>] maxsize layout =<layout like > [attributes]  # vxassist –h datadg maxsize layout=mirror |
| VX | To create a concatenated default volume without specifying ang particular disk | # vxassist –b [-g <dg\_name>]  make <volume\_name> <length>  -b will make the volume immediately available by initializing & starting it  # vxassist –b –g datadg make voldata1  10g |
| VX | To create a concatenated default volume on specified disks | # vxassist –b –g datadg make voldata1  10g  disk01  disk02  Exclude a disk :  # vxassist –b –g datadg make voldata1  10g  !disk05  Exclude disks in a controller :  # vxassist –b –g datadg make voldata1  10g  !ctlr:c2 |
|  | Creating a Basic Mirrored Volume | #  vxassist [-g <dg\_name>]  -b make <volume\_name> <size> layout=mirror |
|  | Create a Vol by vxassist but don’t initiate & start it | Use the option   init=none |
|  |  |  |
| VX | Crerating a volume using a Vxmake description file | Vxmake –d <description\_file> |
| VX | Initialize & start a Volume | # vxvol start <volume>    - enable ,initialize , activae the volume |
| VX | Enable a volume without initializing it to restore data from backup | # vxvol init enable <volume> |
| VX | Then activate the volume | # vxvol init active <volume> |
| VX | To start a DISABLED volume | # vxrecover –s <volume> ,  if u do not specify a name , it will recover all |
|  |  |  |
| VX | Stopping a Enabled volume | # vxvol b [-g <dg\_name>]  stop <volname>   #  vxvol stopall |
| VX | Putting a Mirrored vol in Maintenance mode whose all mirrors are STALE.  Volume will be DETACHED | #  vxvol maint <volume>  # |
|  |  |  |
| VX | To find how much u can grow a Vol  With mentioning the disks | # vxassist maxgrow <volume> , U can not shrink a UFS , but VxFS – grow ,shrink |
| VX FS | To increase a UFS volume ( -b = bg) | # vxresize –F ufs datavol 10g disk10 disk11   ( Mirrored vol needs 2 disks at least) |
| VX  FS | Resizing using vxassist  ( Encapsuled root disk can not be changed) | # vxassist [-b] growto <volume> <new\_size>  # vxassist [-b] growby <volume> <space\_to\_be\_added>  Similary shrinkto & shrinkby if u use Vxfs |
| VX | Removing a STOPPED Volume | # vxassist remove volume <volume> |
| VX | Removing a STOPPED Volume along with plexes , sub disks | # vxedit  -r –f  rm <volume> |
| VX | Moving volume from a VM disk | # vxdiskadm  Menu 7 |

WHAT IS DIFF BETWEEN DISABLED & DETACHED volumes

Increasing Filesystem after increasing Volume number

/usr/lib/fs/vxfs/fsadm -F vxfs -b 28770304  <mount\_point>

The size - 28770304  I got from vxprint –htr <volume\_name>

## VxFS    ( 1 sector = 512 byte)

|  |  |  |
| --- | --- | --- |
| FS | Creating a 12288 Sector VxFS | # mkfs –F vxfs  /dev/vx/**r**dsk/datadg/vol03  12288   ( Can be a non volume raw dev also) |
| FS | Identify a FS type | # fstyp <raw device>  , -v will show lot more info |
| FS | Extending/Shrinking a FS mounted at /data1 to New Size 22000 sectors | # /usr/lib/fs/vxfs/fsadm –F vxfs –b 22000  /data1                                  Using mount point  # /usr/lib/fs/vxfs/fsadm –F vxfs –b 22000 –r /dev/vx/rdsk/datadg/vol1  Using Raw dev |
| FS | Reorganize/Defrag a FS ( Mounted) before shrinking | # /usr/lib/fs/vxfs/fsadm –EeDd /data1  -e extends , -d directories , -E/D report |

You can Extend UFS but can not shrink it