AIX Logical Volume Manager (LVM)

This is a quick and dirty cheat sheet on LVM using AIX, I have highlighted many of the common attributes for each command however this is not an extensive list, make sure you look up the command.

First a quick review on some of the terminology that AIX LVM uses

Examples	What it means
PHYSICAL VOLUME (PV)	Represents a hard disk (hdisk0).
PHYSICAL PARTITION (PP)	The smallest allocation unit in the LVM. All PPs within a VG are the same size, usually 4 or 8 MB.
VOLUME GROUP (VG)	A set of one or more PVs which form a single storage pool. You can define multiple VGs on each AIX system.
LOGICAL VOLUME (LV)	One or more PPs. A file system resides on top of an LV. Only one LV is mapped to a file system. A LV can't span across a VG. Up to 255 LVs in a VG
LOGICAL PARITITION (LP)	One or more PPs. LP represents a mirrored copy of a PP. Up to two copies of a PP can be mirrored resulting in a LP count of three (2 mirrors plus original).
Volume Group Descriptor Area (VGDA)	Information about all the LVs and PVs within a VG. The first 64K of a PV is reserved for this area - defined in <sys/bootrecord.h>. The VGDA consists of • **BOOTRECORD:** - first 512 bytes. Allows the Read Only System (ROS) to boot system • **BAD BLK DIRECTORY** - found in <sys/bddir.h> • **LVM RECORD** - found in <lvmrec.h>
Volume Group Status Area (VGSA)	Information about which PPs that are stale and which PVs are missing within a VG. The LVM and SCSI driver reserves somewhere between 7-10% of the available disk space for LVM maps, etc.
Physical Volume ID (PVID)	The PVID is an amalgamation of the machine's serial number (from the systems EPROMs) and the date that the PVID is being generated. This combination insures theextremely low chance of two disks being created with the same PVID. Finally, when a system is booted, the disk configurator goes and looks at the PVID sitting on each disk platter and then compares that to an entry in ODM. If the entry is found, then the disk is given the hdiskX name that is associated with the ODM entry for the PVID.
Quorum	Quorum is a sort of "sanity" check that LVM uses to resolve possible data confliction and prevent data corruption. Quorum is a method by which 51% or more quorum votes must be available to a volume group before LVM actions can continue. Quorum is issued to a disk in a volume group according to how the disk was created within the volume group. When a volume group consists of one disk, there are two VGDA's on that disk. Thus, this single disk volume group has a quorum vote of 2. When another disk is added to the volume group with an "extendvg", then this new disk gets one VGDA, but the original, first disk still retains the two VGDA's. When the volume group has been extended to three disks, the third disk gets the spare VGDA sitting on the first disk and then each disk has a quorum vote of 1. Every disk after the third disk is automatically given one VGDA, and thus one vote.
Volume Group ID (VGID)	Just as the PVID is a soft serial number for a disk, the VGID is the soft serial number for the volume group. It is this serial number, not the volume group's ascii name, which all low level LVM commands reference. Additionally, it is the basis for the LVIDs created on that VGID.
Logical Volume Control Block (LVCB)	The logical volume control block (lvcb) consists of the first 512 bytes of a logical volume. This area holds important information such as the creation date of the logical volume, information about mirrored copies, and possible mount points in a journaled filesystem.
Logical Volume ID (LVID)	The LVID is the soft serial number used to represent the logical volume to the LVM libraries and low level commands. The LVID is created from the VGID of the volume group, a decimal point, and a number which represents the order which the logical volume was created on the volume group.

Now for the cheet sheet

Directory and Files	
Directories and Files	
Tools	
diagnostic	diag - used to hot swap the disk cfgmgr - used mak sure the new disk is seen # to add new disk from the scsi0 controller cfgmgr -l scsi0
Create/Remove hard disk	cfgmgr -l scsi0 mkdev -c disk -l <pv> rmdev -dl <pv>
Physical Volumes	
display	lspv lspv <pv> (detailed) lspv -l <pv> (list logical volumes) lspv -p <pv> (physical partition usage)
PVID	chdev -l <pv> -a pv=yes chdev -l <pv> -a pv=clear Note: PVID's are automatically added when the disk is placed into a vg

adding	chdev -l <pv> -a pv=yes (new) chpv -v a <pv> (adds back the removed disk)
removing	chpv -v r <pv>
change physical attributes	chpv -a y <pv> (changes allocatable state to YES) chpv -a n <pv> (changes allocatable state to NO)
moving	migratepv <old pv> <new pv>
Volume Groups	
display	lsvg lsvg <vg> (detailed) lsvg -l <vg> (list all logical volumes in goup) lsvg -p <vg> (list all physical volumes in group) lsvg -o (lists all varied on) lsvg -M <vg> (lists assicated disks and state) ## Details volume group info for the hard disk lqueryvg -Atp <pv> lqueryvg -p <disk> -v (Determine the VG ID# on disk) lqueryvg -p <disk> -L (Show all the LV ID#/names in the VG on disk) lqueryvg -p <disk> -P (Show all the PV ID# that reside in the VG on disk)
varyon	varyonvg <vg> varyonvg -f <vg> (force) varyonvg -s <vg> (maintenance mode can use VG commands but lv 's cannot be opened for i/o access) varyoffvg <vg> Note: the varyon command activiates the volume goup which means it is available for use
ODM related	## Determine if the ODM and VGDA are correct (in sync) getlvodm -u <vg> ## tries to resync VGDA, LV control blocks and ODM synclvodm <vg> ## If the message **0516-366 lsvg: Volume group <vg> is locked** is ever seen putlvodm -K `gtlvodm -v <vg>`
creating	mkvg -y <vg> -s <PP size> <pv> mkvg -y datavg -s 4 hdisk1 Note: the PP size will be the size of the physical partition size you want 4MB, 8MB
extending	extendvg <vg> <pv>
reducing	reducevg -d <vg> <pv> ## removes the PVID from the VGDA when a disk has vanished without using the reducevg command reducevg <vg> <PVID>
removing	varyoffvg <vg> exportvg <vg> Note: the export command nukes everything regardingthe volume goup in the ODM and /etc/filesystems
checking	## check to see if underlying disk has grown in size chvg -g <vg> Note: use this command if you are using SAN LUN's that have increased in size
change volume attributes	## auto vary on a volume at system start chvg -a y # Turns on/off quorum checking on a volume group chvg -Q [y|n] <vg>
renaming	varyoffvg <old vg name> lsvg -p <old vg name> (obtain disk names) exportvg <old vg name> import -y <new vg name> <pv> varyonvg <new vg name> mount -a
importing	importvg -y <vg> <pv> importvg <pv> (will use rootvg as default vg)
exporting	varyoffvg <vg> exportvg <vg> Note: if the volume has an active paging space this must be turned off before
Logical Volumes	
display	lslv <lv> lslv -l <lv> (list all physical volumes in logical volume) lslv -m <lv> (list ppartition mapping) ## Display lv control block information

	getlvcb -AT <lv>
creating	mklv <vg> <# of PP's> <pv> mklv -y <lv name> <vg> <# of PP's> <pv> ## Create a mirrored named logical volume mklv -y <lv> -c <copies 2 or 3> <vg> <# of PP's> <pv> ## create a JFSlog logical Volume mklv -y <lv name> -t jfslog <vg> <# of PP's> <pv>
extending	extendlv <lv> <additonal # of PP's> extendlv <lv> <size of volume in B||M|G>
reducing/resizing	see filesystem below
removing	rmlv <lv>
moving	migratepv -l <lv> <old pv> <new pv>
adding a mirror to a non-mirrored volume	mklvcopy -s n <lv> <copies 2 or 3> <pv>
removing a mirror copy from a mirrored volume	rmlvcopy <lv> <copies 1 or 2> rmlvcopy <lv> <copies 1 or 2> <pv> (specified pv) unmirrorvg <vg> <pv>
synchronize logical volume	syncvg -p <pv> syncvg -v <vg> syncvg -l <lv>
mirror any unmirrored volumes	mirrorvg <vg> <pv>
change volume attributes	## Enable the bad-block relocation policy chlv -b [y|n] <lv>
renaming	chlv -n <new lv name> <old lv name>
Miscellaneous	## Initialises an LV for use as an JFSlog logform </dev/lv>
Filesystems	
display	lsfs lsfs -q <fs> (detailed) Note: use the '-q' to see if the logical volume size is bigger than the filesystem size
create	## create new filesystem, -A means to mount after restart crfs -v jfs -d <lv> -m <mountpoint> -A yes ## Create logical volume, filesystem, mountpoint, add entry to /etc/filesystems at the specified size crfs -v jfs2 -g <vg> -m <mountpoint> -a size=<size in 512k blocks|M|G) -A yes Note: there are two types of filesystems jfs and jfs2, jfs2 allows you to decrease the filesystem size , you cannot reduce a jfs filesystem
remove	rmfs <fs> Note: if all filesystems have been removed from a logical volume then the logical volume is removed as well.
resize	chfs -a size=<new size> <fs> chfs -a size=1G /var (specific size, can be used to increase and decrease) chfs -a size=+1G /var (increase by 1GB) chfs -a size=-1GB /var (reduce by 1GB) Note: this will automatically increase or decrease the underlying logical volume as well
freeze/unfreeze	chfs -a freeze=<time in seconds> <fs> chfs -a freeze=off <fs>
split mirrored copy	chfs -a splitcopy=<split copy mountpoint>-a copy=2 <fs> chfs -a splitcopy=/backup -a copy=2 /testfs
change	## Change the mountpoint chfs -m <new mountpoint> <fs> ## Do not mount after a restart chfs -A no <fs> ## Mount read-only chfs -p ro <fs>
mount	mount mount [<fs>|<lv>] mount -a mount all
defrag	defragfs -q <fs> (report defrag status) defragfs -r <fs> (runs in report only mode - no action) defragfs <fs> (actually defrags the filesystem)

checking and repairing	fsck [-y|-n] <fs> (check a filesystem) fsck -p <fs> (restores primary superblock from backup copy if corrupt)
Miscellaneous	
Complete VG, LV and FS with mirroring example	## Create the volume group mkvg -s 256 -y datavg hdisk2 ## Create the jfs2 log logical volume and initialize it this for the volume group mklv -t jfs2log -y dataloglv datavg 1 logform /dev/dataloglv ## Create the logical volume mklv -t jfs2 -y data01lv datavg 8 ## Create the filesystems that will use the logical volume crfs -v jfs -d data01lv -m /data01 -A yes ## Add an additional hard disk to the volume group extendvg datavg hdisk3 ## Now mirror both the volume group log logical volume and the logical volume mklvcopy dataloglv 2 mklvcopy data01lv 2 ## Make sure everything is sync'ed both the log and the logical volume syncvg -v datavg ## Make sure everything is OK lsvg -l datavg ## a quick way to perform the above in two steps mklv -c 2 -t jfs2 -y data02lv datavg 8 crfs -v jfs -d data02lv -m /data02 -A yes ## mount everything and check mount -a
Replaced failed mirror drive	## break the mirror (two ways to do this) rmlvcopy <lv name> 1 <broken disk> unmirrorvg <lv> <broken pv > ## remove the disk from the vg reducevg <vgname> <broken pv > ## remove the hdisk from ODM rmdev -dl <broken pv> ## physically replace the disk diag -> function select -> task selection -> hot plug task -> scsi and scsi raid hot plug manager -> replace/remove a device attached to an scsi hot swap enclosure device -> select disk and follow instructions ## configure new disk an check the new number (hopefully the same) cfgmgr -v lsdev -Cc <pv> ## add back to volume group extendvg <vg> <pv> ## create mirror (two ways to do this) mklvcopy <lv> 2 <pv> mirrorvg <lv> ## sync mirror syncvg -l <lv> ## If this is the rootvg there are additonal steps to take bosboot -ad /dev/<pv> bootlist -m normal <pv> <pv> bootlist -m normal -o
Accidently remove a mirrored disk or SAN LUN disappeared off the network	## This procedure places back a mirror disk that you have accidently pulled or that a SAN LUN disappeared off the network ## and its states is classed as "missing" ## see that the disk is in a missing state (see PV state column), also see stale volumes lsvg -p <vg> lsvg -M <vg> ## To make the disk active again we use the varyonvg command varyonvg <vg> ## see that the disk is in a active state (see PV state column) lsvg -p <vg> ## Now re-sync the volumes in that volume group syncvg -v <vg>

	## Make sure that no volumes are stale lsvg -M <vg> ## Determine if the ODM and VGDA are correct (in sync) getlvodm -u <vg>

For other LVM's and Array utilities see my LVM central page