

LVM New Features in HP-UX 11i v3



Abstract	3
New functionality	4
Support for Agile View of Mass Storage	4
Multipathing and Alternate Links (PVlinks)	4
Dynamic LUN Expansion (DLE)	4
Modification of Volume Group Settings	8
Modification of Physical Volume Type (Boot/Non-Boot)	8
SLVM Online Reconfiguration	9
LVM Device Online Replacement (OLR).....	9
Volume Group Quiesce and Resume	9
Boot Resiliency.....	10
Striped and Mirrored Logical Volumes.....	10
Better Coexistence with Other Disk Users.....	14
Better Utilization of Disk Space – No Bad Block Reserve Area	14
Elimination of maxvgs Tunable	15
Proactive Polling of Alternate links	16
Version 2.x Volume Groups.....	16
Boot Disk Configuration Self Healing.....	16
Performance Improvements.....	17
Mirror Write Cache (MWC) Enhancements.....	17
LVM Support for Large I/O Sizes.....	17
Parallel Resynchronization of Stale Extents.....	18
Parallel Open of Physical Volumes.....	18
Increased Limits.....	19
Maximum Logical Volume Size Increased up to 16TB	19
Increased Limits for Version 2.x Volume Groups.....	19
Usability Enhancements	20
Compact, Parsable Command Output	20
pvdisplay – Displays user data offset and checks if disk is under HP-UX LVM control.....	21
vgscan - Scans Faster, Per Volume Group Scanning and Supports Persistent DSF	22
vgcfdgrestore – Enhanced Listing of Backup File Contents	25
Error Management Technology (EMT).....	26
Long Host Name Support for Display Commands.....	26
lvmadm Command	27
vgremove – Removal of Volume Group Device Files	28
vgdisplay – Displays More Information	28

Auto Creation of Volume Group File	30
pvmove Enhancements	30
Miscellaneous.....	32
Commands enhanced to prevent misconfigurations through alternate links	32
Mirror Disk Installation No Longer Requires a Reboot.....	32
Glossary	33
For More Information	35
Call to Action.....	35

Abstract

In HP-UX 11i v3 (11 .31), LVM delivers significant performance, scalability, usability, and availability enhancements. After the initial release of HP-UX 11i v3 in February 2007, there have been three updates of HP-UX 11i v3: September 2007, March 2008, and September 2008. This whitepaper lists all the new LVM features in the initial release and the three updates of HP-UX 11i v3.

Some features have their own whitepaper; this document only summarizes them. For more information, see the referenced documents in the [For More Information](#) section. Other features are presented in more detail, including their use and benefits.

The document is intended for system administrators, operators, and customers who want to use and know about new LVM features in HP-UX 11i v3.

New functionality

Support for Agile View of Mass Storage

HP-UX 11i v3 introduces a new representation of mass storage devices called the **agile view**. In this representation, the device special file (DSF) name for each disk no longer contains path (or link) information. The multipathed disk has a single **persistent DSF** regardless of the number of physical paths to it. The **legacy view**, represented by the **legacy DSF**, continues to exist. You can use both the DSF types to access a given mass storage device independently and both can coexist on a system. For more information, see the [The Next Generation Mass Storage Stack](#) whitepaper.

Wherever applicable, LVM configuration commands support both DSF naming models.

LVM enables you to configure volume groups with all persistent DSFs, all legacy DSFs, or a combination of persistent DSFs and legacy DSFs.

HP recommends the use of persistent DSFs for LVM configurations and encourages configuring new volume groups using persistent DSFs. To fully utilize all the capabilities of the new mass storage stack, HP recommends migrating volume groups from legacy DSFs to persistent DSFs. HP provides `/usr/contrib/bin/vgdsf` to facilitate this migration. The script works for both root and non-root volume groups. For more information, see the [LVM Migration from Legacy to Agile Naming Model](#) whitepaper.

Multipathing and Alternate Links (PVLlinks)

Management of multipathed devices is available outside of LVM using the next generation mass storage stack. Agile addressing creates a single persistent DSF for each mass storage device regardless of the number of hardware paths to the disk. The mass storage stack in HP-UX 11i v3 uses that agility to provide transparent multipathing. For more information, see the [The Next Generation Mass Storage Stack](#) whitepaper.

The LVM Alternate Link functionality is now redundant, but this functionality is still supported with legacy DSFs. Alternate links behave as they did in previous releases when the mass storage stack native multipathing feature is disabled with the `scsimgr` command.

HP recommends converting volume groups with alternate links to use native multipathing by the use of persistent DSFs. The `/usr/contrib/bin/vgdsf` script, `vgscan -N` command, or `vgimport -s -N` commands perform this conversion. For more information, see the [LVM Migration from Legacy to Agile Naming Model](#) whitepaper.

Dynamic LUN Expansion (DLE)

Some disk arrays allow the dynamic resizing of their LUNs. With HP-UX 11i v3, LVM detects and handles physical volume size changes on invocation of the `vgmodify` command. This command is supported for Version 1.0 volume groups only. For more information on different versions of volume groups that LVM supports, see the ["Version 2.x Volume Groups"](#) section.

vgmodify(1M)

The `vgmodify` command provides a method to alter the attributes of a physical volume and volume group after `pvcreeate` and `vgcreate` operations, respectively. You must run the `vgmodify` command to update the LVM configuration to reflect any change to the physical volume size.

Prior to September 2008 release of HP-UX 11i v3, you can alter the attributes of a physical volume or a volume group using the `vgmodify` command only when the volume group is deactivated. Starting with the September 2008 release of HP-UX 11i v3, LVM supports handling dynamic LUN expansion of physical volumes using the `vgmodify` command when the volume group is active. This is referred to as online DLE. To support online DLE,

LVM introduces two new options: *-a* (to indicate an online operation) and *-E* (to handle a DLE operation explicitly).

For more information, see `vgmodify(1M)` and the - [LVM Volume Group Dynamic LUN expansion \(DLE\)/vgmodify](#) whitepaper.

pvmove(1M)

The *pvmove* command was enhanced in the February 2007 release of HP-UX 11i v3 to allow relocation of just the first extent of the physical volume. The *vgmodify* command uses this additional space to expand the LVM's on-disk configuration information.

The *pvmove* command honors the existing allocation policy of a logical volume containing the extent that is considered for relocation.

For more information, see `pvmove(1M)`

Examples

- To relocate the first data extent to any free extent within the same physical volume, enter:
`# pvmove /dev/disk/disk10:0 /dev/disk/disk10`
- To relocate the first data extent to any free extent in the volume group, enter:
`# pvmove /dev/dsk/c1t0d0:0`
- To find a physical volume that has free space, use `pvdiskdisplay(1M)`. You can then relocate the first user extent to that physical volume using *pvmove*:

```
# pvdiskdisplay /dev/disk/disk22

--- Physical volumes ---
PV Name           /dev/disk/disk22
VG Name           /dev/vgname
PV Status          available
Allocatable        yes
VGDA               2
Cur LV            5
PE Size (Mbytes)   4
Total PE           1279
Free PE           779
Allocated PE       500
Stale PE           0
IO Timeout (Seconds) default
Autoswitch         On
```

To relocate first data extent from the source physical volume to the desired destination physical volume that was identified in the last step, enter:

```
# pvmove /dev/disk/disk10:0 /dev/disk/disk22
```

Note: Relocation of the first data extent fails if it violates the strict mirror allocation policy. So identify a suitable physical volume that does not hold any mirror copies of the same extent which is considered for relocation.

- To relocate the first data extent of a physical volume when the *vgmodify* command reports insufficient space for expanding the LVM configuration data on the disk. This happens when the user tries to modify the physical volume settings using the *vgmodify* command.

Consider a volume group with one physical volume and one logical volume as follows:

```
# vgdisplay -v /dev/vgdatabase
```

```
--- Volume groups ---
```

```
VG Name                /dev/vgdatabase
VG Write Access        read/write
VG Status              available
Max LV                 255
Cur LV                1
Open LV                1
Max PV                 16
Cur PV                1
Act PV                 1
Max PE per PV         2559
VGDA                   2
PE Size (Mbytes)      4
Total PE               2559
Alloc PE               10
Free PE                2549
Total PVG              0
Total Spare PVs       0
Total Spare PVs in use 0
```

```
--- Logical volumes ---
```

```
LV Name                /dev/vgtest/lv011
LV Status              available/syncd
LV Size (Mbytes)      40
Current LE             10
Allocated PE           10
Used PV                1
```

```
--- Physical volumes ---
```

```
PV Name                /dev/dsk/c11t0d5
PV Status              available
Total PE               2559
Free PE                2549
Autoswitch             On
```

The *vgmodify* command can report optimized volume group settings to adjust the number of extents and physical volume upwards, where possible, to make full use of the space reserved on each physical volume for the LVM configuration data:

```
# vgmodify -o -r /dev/vgtest
```

```
Current Volume Group settings:
```

```
Max LV                255
Max PV                 16
Max PE per PV         2559
PE Size (Mbytes)      4
VGRA Size (Kbytes)    400
```

```
New configuration requires "max_pes" are increased from 2559 to 6652
```

```
The current and new Volume Group parameters differ.
```

```
An update to the Volume Group IS required
```

```
New Volume Group settings:
```

```
Max LV                255
Max PV                 16
Max PE per PV         6652
PE Size (Mbytes)      4
VGRA Size (Kbytes)    896
```

```
Review complete. Volume group not modified
```

The previous output shows that you can change the "Max PE per PV" value to 6652 with the value of "Max PV" retained at 16. To increase the value of "Max PV" (with the current available space for LVM configuration data), it might be necessary to reduce the value of "Max PE per PV". Use the *-t* option of the *vgmodify* command to check if the "Max PV" value can be increased by changing the value of "Max PE per PV". Alternatively, you can relocate the first data extent to create more space for LVM configuration data and increase the "Max PV" value without reducing the "Max PE per PV" value. To achieve this, use the *-n* option of the *vgmodify* command.

Any attempt to increase the maximum physical volume setting in the volume group to a value greater than 16 results in a failure as follows:

```
# vgchange -a n /dev/vgtest
```

```
Volume group "vgtest" has been successfully changed.
```

```
# vgmodify -n -p 64 /dev/vgtest
```

```
Current Volume Group settings:
```

Max LV	255
Max PV	16
Max PE per PV	2559
PE Size (Mbytes)	4
VGRA Size (Kbytes)	400

```
vgmodify: This operation can only be completed if PE number zero on  
"/dev/rdisk/c11t0d5" is freed
```

Note that the last output message indicates you must free the first data extent. Use the *pvmove* command to relocate the first data extent as follows:

```
# vgchange -a y /dev/vgtest
```

```
Activated volume group
```

```
Volume group "/dev/vgtest" has been successfully changed.
```

```
# pvmove /dev/dsk/c11t0d5:0
```

```
Transferring logical extents of logical volume "/dev/vgtest/lvol1"...
```

```
Physical volume "/dev/dsk/c11t0d5" has been successfully moved.
```

```
Volume Group configuration for /dev/vgtest has been saved in  
/etc/lvmconf/vgtest.conf
```

Use the *vgmodify* command to increase the maximum physical volume setting. This time it succeeds:

```
# vgchange -a n /dev/vgtest
```

```
Volume group "vgtest" has been successfully changed.
```

```
# vgmodify -n -p 64 /dev/vgtest
```

```
Current Volume Group settings:
```

Max LV	255
Max PV	16
Max PE per PV	2559
PE Size (Mbytes)	4
VGRA Size (Kbytes)	400

```
The current and new Volume Group parameters differ.  
An update to the Volume Group IS required
```

```
New Volume Group settings:
```

Max LV	255
Max PV	64
Max PE per PV	2559
PE Size (Mbytes)	4
VGRA Size (Kbytes)	1488

```
New Volume Group configuration for "vgtest" has been saved in  
"/etc/lvmconf/vgtest.conf"
```

```
Old Volume Group configuration for "vgtest" has been saved in  
"/etc/lvmconf/vgtest.conf.old"
```

```
Starting the modification by writing to all Physical Volumes
Applying the configuration to all Physical Volumes from "/etc/lvmconf/vgtest.conf"
Completed the modification process.
New Volume Group configuration for "vgtest" has been saved in
"/etc/lvmconf/vgtest.conf.old"
Volume group "vgtest" has been successfully changed.
```

Now issuing *vgdisplay* on the volume group shows the modified values (in this case the maximum physical volume) for the volume group as follows:

```
# vgchange -a y vgtest
Activated volume group
Volume group "vgtest" has been successfully changed.

# vgdisplay /dev/vgtest
VG Name                /dev/vgtest
VG Write Access        read/write
VG Status              available
Max LV                 255
Cur LV                1
Open LV                1
Max PV                64
Cur PV                1
Act PV                 1
Max PE per PV         2559
VGDA                   2
PE Size (Mbytes)      4
Total PE               2558
Alloc PE               10
Free PE                2548
Total PVG              0
Total Spare PVs       0
Total Spare PVs in use 0
```

Modification of Volume Group Settings

When an LVM volume group is created, several configuration parameters are set (for example, *max_pe*, *max_pv*, *max_lv*). The new *vgmodify* command enables you to change these configuration parameters on an existing volume group, which avoids having to migrate user data.

The *vgmodify* command can alter the following three configuration parameter set via *vgcreate*:

- The maximum number of physical extents that can be allocated per physical volume (*max_pe* setting set by *vgcreate -e*).
- The maximum number of physical volumes that the volume group can contain (*max_pv* setting set by *vgcreate -p*).
- The maximum number of logical volumes that the volume group can contain (*max_lv* setting set by *vgcreate -l*).

The *vgmodify* command displays the possible volume group *max_pe* and *max_pv* settings for this volume group to help optimize an existing volume group configuration.

For more information, see *vgmodify(1M)* and the [LVM Volume Group Dynamic LUN expansion \(DLE\)/vgmodify](#) whitepaper.

Modification of Physical Volume Type (Boot/Non-Boot)

When initializing a physical volume for LVM, *pvcreate* assigns a type to it, either boot or non-boot (Note that bootable physical volumes are not supported in Version 2.x volume groups).

The *vgmodify* command enables you to change a physical volume type from boot to non-boot or vice versa.

See *vgmodify(1M)* and *pvcreate(1M)* for the *-B* option. Also see the [LVM Volume Group Dynamic LUN expansion \(DLE\)/vgmodify](#) whitepaper.

Note that making a physical volume non-bootable increases the space available on that device for LVM configuration data. However, to take advantage of the additional space, you must mark every disk in a volume group non-bootable.

SLVM Online Reconfiguration

Starting with the HP-UX 11i v3 September 2009 release, you can change the configuration of an active shared volume group in a cluster without deactivating the volume group on any node. Applications stay available on all nodes sharing the volume group. This applies to Version 2.1 and higher volume groups.

For Version 1.0 and 2.0 volume groups, you can use the *vgchange -x* option, the SLVM SNOR feature to change the configuration of a shared volume group, while keeping it active only on a single node. Using this procedure, applications on at least one node remain available during the volume group reconfiguration.

For more information, see the SLVM Online Reconfiguration whitepaper.

LVM Device Online Replacement (OLR)

The LVM Online Disk Replacement (OLR) feature provides the following new methods for replacing or isolating path components or LVM disks within an active volume group:

- Using *-n* and *-N* options with *pvchange* command, you can detach a specific path or all paths to a physical volume, respectively. LVM OLR enables you to follow a simpler procedure for replacing disks in an active volume group. The procedure does not require deactivating the volume group, modifying the volume group configuration, or moving any user data.
- Using LVM OLR, you can easily isolate troublesome paths or disks to allow running diagnostics against them.

In HP-UX 11i v3, the option of detaching an entire physical volume using *pvchange -a N* command, in order to perform an online disk replacement, is still supported. The behavior is the same for both legacy and persistent DSFs and is compatible with previous releases.

Unless native multipathing is disabled and only legacy DSFs are configured for the volume group, the *pvchange -a n* command does not stop I/O operations for that path as they did in earlier releases. Instead, use the *scsimgr* command with the *disable* option to disable physical volume paths. For more information, see *scsimgr(1M)*.

For more information, see the [LVM Online Disk Replacement \(LVM OLR\)](#) whitepaper.

Volume Group Quiesce and Resume

The LVM volume group Quiesce/Resume feature allows quiescing I/O operations to the disks in a volume group to facilitate creating a consistent snapshot of an otherwise active volume group for backup purposes. This feature works with backup management and disk array management software to enable them to create a consistent snapshot of the disks that make up an LVM volume group.

The Quiesce/Resume feature prevents the disk images from changing and allows a snapshot of the disks to be taken without having to unmount or close the open logical volumes and deactivate the volume group.

The *vgchange* command provides new options *-Q* and *-R* to allow quiescing the volume group prior to creating a snapshot and to resume the volume group afterward. Optionally, both reads and writes or just writes to the volume group can be quiesced.

For more information, see the [LVM Volume Group Quiesce/Resume](#) whitepaper.

Boot Resiliency

Root volume group scanning is a new LVM feature in HP-UX 11i v3. The feature can prevent boot failures that can occur on prior HP-UX releases.

During boot, the root volume group activation can fail if the LVM boot configuration information is incorrect or out-of-date with the systems current I/O configuration. Two possible causes are:

- The root volume group is configured using legacy DSFs representing the devices in a Storage Area Network(SAN) and the SAN is reconfigured such that DSFs of the devices changed.
- The root disk is relocated to a different slot such that the DSF name changes.

With the new root volume group scanning, LVM automatically handles such situations; LVM now scans all the disks to identify the ones belonging to the root volume group and retries activation. If the activation succeeds, it is likely that the LVM in-memory boot configuration information for the root volume group is out of sync with the DSF in the `/etc/lvmtab` for the root volume group. To assist recovery in this case, the LVM driver prints a warning message to the console and logs into `/var/adm/syslog/syslog.log` to the effect as follows:

```
"LVM: WARNING: Root VG activation required a scan. The PV information in the on-disk
BDRA may be out-of-date from the system's current IO configuration. To update the
on-disk BDRA, first update /etc/lvmtab using vgscan(1M), then update the on-disk
BDRA using lvmboot(1M).
```

For example, if the root VG name is `/dev/vg00`:

```
vgscan -k -f /dev/vg00
lvmboot -R /dev/vg00"
```

In case some physical volumes in the root volume group are not available but the quorum is met, no root volume group scan is performed. Also, during a single user mode boot with `-is` or maintenance mode boot with `-lm`, root volume group scanning is skipped.

Note: This feature is enhanced further in the September 2008 release of HP-UX 11i v3. For more information, see the "[Boot Disk Configuration Self Healing](#)" section.

Striped and Mirrored Logical Volumes

HP-UX LVM now introduces support for striped and mirrored logical volumes at a smaller granularity than the extent size (4KB is the smallest possible stripe size).

RAID 1+0 and RAID 0+1

RAID 0, commonly referred to as *striping*, refers to the segmentation of logical sequences of data across disks. RAID 1, commonly referred to as *mirroring*, refers to creating exact copies of logical sequences of data. When implemented in a device hardware, RAID 10 (or RAID 1+0) and RAID 01 (or RAID 0+1) are nested RAID levels. The difference between RAID 0+1 and RAID 1+0 is the location of each RAID system — RAID 0+1 is a mirror of stripes whereas RAID 1+0 is a stripe of mirrors. Figure 1 shows the RAID 10 and RAID 01 configurations (A1, A2...Ax are stripe chunks of a logical volume).

With a hardware-based RAID 10 configuration, I/O operations are striped first then each strip is mirrored. With hardware-based RAID 01, I/Os are mirrored first then striped. RAID 10 and RAID 01 can have the same physical disk layout.

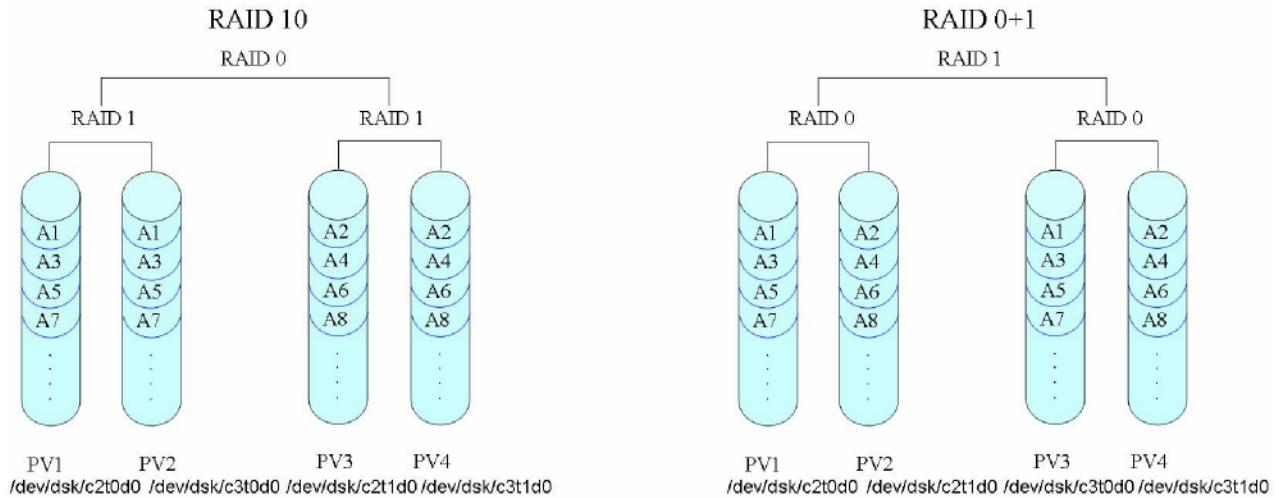


Figure 1: RAID 1+0 and RAID 0+1

The advantages of hardware-based RAID 10 over RAID 01:

- When one disk fails and is replaced, only the amount of data on this disk needs to be copied or re-synchronized.
- RAID 10 is more tolerant to multiple disk failures before data becomes unavailable.

The advantages of hardware-based RAID 01 over RAID 10:

- Simpler to configure striped volumes and then extend mirroring.
- Able to split the mirror copy and have two usable volume sets.

LVM Implementation of RAID levels in HP-UX

LVM implementation of RAID management is different from the hardware based solutions because it does not nest the RAID levels, but processes them simultaneously. Typically with hardware solutions, you create a LUN with a RAID level and the RAID functions are stacked. LVM provides more flexibility on how logical volumes are created amongst a set of disks as compared to hardware solutions.

LVM allocates the physical extents for striped and mirrored logical volumes in sets of stripe width multiplied by the number of copies of the data. For example, if the logical volume is 1-way mirrored and striped across two disks, extents are allocated to the logical volume, four at a time. LVM enforces that the physical extents of a single set are from different physical volumes. Within this set, the logical extents are striped and mirrored to obtain the data layout displayed in Figure 1.

Striping and mirroring in LVM combines the advantages of the hardware implementation of RAID 1+0 and RAID 0+1, and provides the following benefits:

- Better write performance. Write operations take place in parallel and each physical write operation is directed to a different physical volume.
- Excellent performance for read. Even in the case where several disks are out of service, the read of a stripe can be done in parallel on different physical volumes with one I/O operation per physical volume.
- High availability of data. With multiple copies of the user data residing on different physical volumes, LVM avoids single point of failure ensuring high availability.

Configuration

You can create a new striped and mirrored logical volume from scratch, or extend (add mirror to) an existing striped logical volume. With the use of physical volume groups (PVGs), mirroring and striping can be directed to specific physical volumes. Without the constraint created by the nesting of levels in hardware based RAID solutions, LVM is combining the striping and mirroring processing, which allows LVM to provide the best of both RAID 10 and RAID 01.

To create a pure RAID 10, ensure that the physical volumes have enough extents in terms of size and stripe width to accommodate the logical volume. Otherwise, LVM optimizes the extent allocation to enable you to create a striped and mirrored logical volume as long as there are enough free extents in the volume group. The logical volume layout then uses a slight variation of RAID 10 and all the benefits of RAID 10 are retained. For example, Figure 2 shows a volume group with seven physical volumes and a logical volume striped across three disks with one mirror copy.

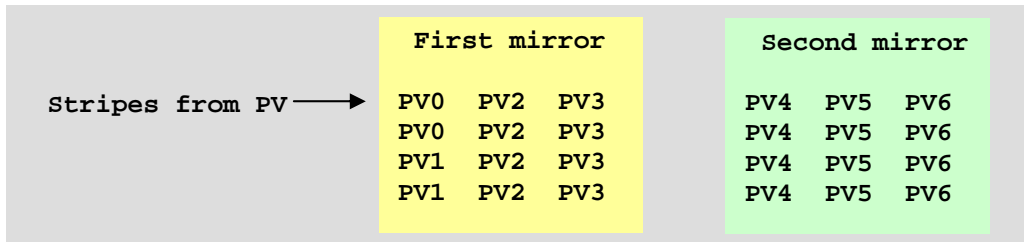


Figure 2: Example of LVM's implementation of RAID 10 variation.

If the first physical volume used in the creation has enough extents (size of LV/3) to create RAID 10, only six physical volumes are used, resulting in a strict RAID 10 configuration. But if one of the physical volumes, PV0 does not have enough free extents; LVM uses another physical volume, PV1, to complete the operation. PV0 was short by two extents so two extents were allocated from PV1. LVM created the logical volume using seven physical volumes resulting in a slight variation of RAID 10.

You can extend your existing striped logical volumes to a striped and mirrored configuration easily, using the *lvextend* command to add mirrors. You cannot convert existing mirrored logical volumes to a striped and mirrored configuration. You can create any new logical volumes with striped and mirrored configuration as explained in the 'Examples' section.

LVM allows the splitting and merging of striped and mirrored logical volumes. These logical volumes are also supported in a clustered environment with shared and in exclusive mode activation.

Examples

Use any of the following procedures to create a striped and mirrored logical volume:

- To create a logical volume of size 90MB striped across two physical volumes with one mirror copy and stripe size of 64 KB, enter.

```
# lvcreate -L 90 -i 2 -I 64 -m 1 /dev/vgtest
```

Note: Striping with mirroring always uses strict allocation policies where copies of data do not exist on the same physical disk. This results in a configuration similar to the RAID 01 as illustrated in Figure1.

- To create a logical volume of size 90MB striped across two physical volumes with one mirror copy and stripe size of 64 KB and to create the mirror copies on specific disks (configuration equivalent to RAID 01 as illustrated in Figure1, enter)

```
# lvcreate -L 90 -i 2 -I 64 /dev/vgtest
```

```
# lvextend -m 1 /dev/vgtest/lvol1 /dev/dsk/c3t0d0 /dev/dsk/c3t1d0
```

```
Contents of /etc/lvm/vg
VG /dev/vgtest
PVG PVG0
/dev/dsk/c2t0d0
/dev/dsk/c2t1d0
PVG PVG1
/dev/dsk/c3t0d0
/dev/dsk/c3t1d0
```

- Adding a mirror to existing striped logical volumes.

The existing logical volume has the striped property as shown by the *lvdisplay* output:

```
# lvdisplay /dev/vgtest/lvoll

--- Logical volumes ---
LV Name                /dev/vgtest/lvoll
VG Name                /dev/vgtest
LV Permission          read/write
LV Status              available/syncd
Mirror copies       0
Consistency Recovery   MWC
Schedule           striped
LV Size (Mbytes)      1024
Current LE             256
Allocated PE           256
Stripes             2
Stripe Size (Kbytes) 64
Bad block              on
Allocation              strict
IO Timeout (Seconds)  default
```

To get a striped and mirrored configuration, extend the logical volume as follows:

```
# lvextend -m 1 /dev/vgtest/lvoll
```

Note that the volume group must have enough physical volumes and extents to accommodate the mirror copy.

Now the *lvdisplay* output shows the logical volume is striped and mirrored:

```
# lvdisplay /dev/vgtest/lvoll

--- Logical volumes ---
LV Name                /dev/vgtest/lvoll
VG Name                /dev/vgtest
LV Permission          read/write
LV Status              available/syncd
Mirror copies       1
Consistency Recovery   MWC
Schedule           striped
LV Size (Mbytes)      1024
Current LE             256
Allocated PE           256
Stripes             2
Stripe Size (Kbytes) 64
Bad block              on
Allocation              strict
IO Timeout (Seconds)  default
```

Compatibility Note

Releases prior to HP-UX 11i v3 only support striped or mirrored logical volumes and do not support combination of striped and mirrored logical volumes. If a logical volume using simultaneous mirroring and striping is created on HP-UX 11i v3, attempts to import or activate its associated volume group fails on a previous HP-UX release.

To import the volume group with striped and mirrored logical volume to releases prior to HP-UX 11i v3, you must remove the incompatible logical volumes or reduce them to no mirrors.

NOTE: Striped and mirrored logical volume is supported on HP-UX 11i v2 with the PHCO_36744, PHKL_36745 and PHCO_36746 patches. With these patches or superseding patches installed on a HP-UX 11i v2 system, you can successfully import and activate a volume group with striped and mirrored logical volume created on HP-UX 11i v3.

Better Coexistence with Other Disk Users

LVM coexists better with other disk users. The LVM commands *pvcreate* and *vgcfgrestore* now check if their target device is in use by other subsystems.

LVM has a provision to disown or unmark a LVM disk. This is facilitated by the *pvremove* command. For more information, see the *pvremove(1M)* man page.

Note: *pvcreate* with the force option *-f* overrides this check and takes ownership of the disk, overwriting any other subsystems data. HP recommends NOT using the *-f* option by default in any of your scripts or during initializing a physical volume using *pvcreate* command.

Example

Coexistence with Oracle ASM disk – If the disk `/dev/dsk/c3t1d1` is marked for use by Oracle ASM, an initialization operation on this disk identifies that the disk belongs to some other subsystem as follows:

```
# pvcreate /dev/rdisk/c3t1d1
pvcreate: Could not perform LVM operation on Oracle ASM disk "/dev/rdisk/c3t1d1"
```

To disown or unmark an LVM disk that is not part of any volume group, use the *pvremove* command as follows:

```
# pvremove /dev/rdisk/disk26
The physical volume associated with "/dev/rdisk/disk26" has been removed.
```

Better Utilization of Disk Space – No Bad Block Reserve Area

Software bad block relocation refers to capability of LVM to handle I/O errors on disk blocks that are bad. In previous releases of HP-UX, when a media failure occurs (detection of a bad block of data on disk), LVM marks the failed block in the Bad Block Directory and attempts to relocate the block to a new location in the Bad Block Reserve Area (BBRA) on the disk.

The BBRA was created when the disk was initialized using the *pvcreate* command.

All modern disks now offer hardware based bad sector relocation and do not trigger the LVM software relocation. With HP-UX 11i v3, LVM does not reserve a BBRA at the end of disk to allocate more space for user data.

Compatibility Note

In HP-UX 11i v3, LVM is compatible with physical volumes that already have entries in the bad block directory. For example, a disk device is configured for LVM on HP-UX 11i v2 and this physical volume has bad blocks relocated to the BBRA. On importing such a physical volume onto a HP-UX 11i v3 release, LVM continues to support this physical volume and can read the relocated blocks from the BBRA.

Elimination of maxvgs Tunable

This static tunable specified the maximum number of LVM volume groups in the range 1 to 256, which could be created or activated on the system. With HP-UX 11i v3 release, the maxvgs tunable was removed. Users can now create up to 256 Version 1.0 volume groups and 2048 Version 2.x volume groups (the combined number of Version 2.0 and Version 2.1 volume groups cannot exceed 2048) in a system without requiring setting this kernel tunable, thus avoiding a reboot.

For more information on different versions of volume groups that LVM supports, see the "[Version 2.x Volume Groups](#)" section and the "[LVM Version 2.0 Volume Groups in HP-UX 11i v3](#)" whitepaper.

Note: You must change scripts that use or modify maxvgs tunable for HP-UX 11i v3.

Example

On a HP-UX 11i v2 system:

```
# kctune maxvgs
Tunable Value Expression
maxvgs      10 Default
```

Modifying the maxvgs tunable on a HP-UX 11i v2 release results in a reboot:

```
# kctune maxvgs=64
```

```
NOTE: The configuration being loaded contains the following change(s)
that cannot be applied immediately and which will be held for
the next boot:
```

```
-- The tunable maxvgs cannot be changed in a dynamic fashion.
```

```
WARNING: The automatic 'backup' configuration currently contains the
configuration that was in use before the last reboot of this
system.
```

```
==> Do you wish to update it to contain the current configuration
before making the requested change? y
```

```
* The automatic 'backup' configuration has been updated.
```

```
* The requested changes have been saved, and will take effect at
next boot.
```

```
Tunable Value Expression
maxvgs (now) 10 Default
(max boot) 64 64
```

On a HP-UX 11i v3 system, no reboot is required as the maxvgs tunable is removed:

```
# kctune maxvgs
ERROR: There is no tunable named 'maxvgs'.
```

For more information, see maxvgs(5).

Proactive Polling of Alternate links

With September 2007 release of HP-UX 11i v3, LVM supports a new *-p* command line option in the *pvchange* command, which controls the way LVM monitors alternate links to a multipathed physical volume.

By default, LVM proactively monitors the health of all configured links. If auto-switchback is enabled (by default), this polling helps LVM detect and switch to a better link when it becomes available.

If proactive polling of alternate links is undesirable, you can disable polling using *pvchange -p n*. For more information, see *pvchange(1M)*.

Version 2.x Volume Groups

With the March 2008 release of HP-UX 11i v3, LVM supports a new version of volume group- Version 2.0. Version 1.0 is the version supported on all current and previous versions of HP-UX 11i.

A Version 2.0 volume group is a volume group whose metadata layout is different from that of Version 1.0 volume groups. Version 2.0 enables the configuration of larger volume groups (up to 512), logical volumes (up to 511), physical volumes (up to 511), and several other volume group parameters. You can manage a Version 2.0 volume group the same way as a Version 1.0 volume group, using the same user interface. But features such as sparing, bad block relocation, cluster lock, boot, dump, and primary swap are not supported on Version 2.0 volume groups. Also, *vgmodify*, *pvck*, *lvlnboot*, and *lvrmboot* commands are not supported on Version 2.0 volume groups.

With the September 2008 release of HP-UX 11i v3, LVM supports another version of volume group- version 2.1. You can also manage a Version 2.1 volume group the same way as a Version 1.0 volume group, using the same user interface. Version 2.1 is similar to version 2.0, but allows configuration of much greater number of volume groups (up to 2048), physical volumes (up to 2048), and logical volumes (up to 2047). With this release, cluster lock is supported on Version 2.x volume groups.

For more information on Version 2.x volume groups, see the [LVM Version 2.0 Volume Groups in HP-UX 11i v3](#) whitepaper. For more information on the supported limits of Version 1.0 and Version 2.x volume groups, see the [LVM Limits White Paper](#).

Boot Disk Configuration Self Healing

With the September 2008 release of HP-UX 11i v3, LVM supports self healing of boot disk configurations. If the device special file for a LVM boot disk changes because its hardware path changed, LVM scans for the boot disk during the boot process. After successfully booting, LVM automatically updates the */etc/lvmtab* LVM configuration file and the LVM metadata on each bootable disk in the root volume group with the new boot path. Future system boots will not need to scan. This behavior is configurable, and is enabled by default.

For more information, see *lvmdm(1M)* and *lvm(7)*.

Performance Improvements

Mirror Write Cache (MWC) Enhancements

The MWC allows a fast resynchronization of data following a system crash or failure.

The MWC keeps track of where I/O writes are occurring on the volume group and periodically records this activity in an on-disk data structure. An extra disk write is required for every mirrored write not already recorded on the physical volume. Upon system reboot after crash, the operating system uses the MWC to resynchronize inconsistent data blocks quickly.

In HP-UX 11i v3, the MWC is larger in size than in previous releases. This leads to a better logical volume I/O performance by allowing more concurrent writes. MWC has also been enhanced to support large I/O sizes.

To enable or disable MWC, use the *-M y* or *-M n* option with *lvcreate* or *lvchange* command, respectively. For more information on MWC, see *lvcreate(1M)* and *lvchange(1M)*.

Logical volumes belonging to shared volume groups (those activated with *vgchange -a s*) of version 1.0 and 2.0 must have the consistency recovery set to NOMWC or NONE. Versions 1.0 and 2.0 do not support MWC for logical volumes belonging to shared volume groups.

With the September 2008 release of HP-UX 11i v3, LVM supports MWC for logical volumes belonging to version 2.1 shared volume groups. This ensures faster recovery following a system crash.

Compatibility Note:

When a volume group containing a logical volume (created on pre-11i v3 releases) using the MWC is activated on HP-UX 11i v3, its MWC format is converted. Any subsequent activation of this volume group on previous releases do not recognize the new MWC format and LVM performs a full resynchronization of the entire logical volume. But if the logical volumes are marked "NOMWC" and the logical volume is synced, activation of the volume group on previous releases avoids full resynchronization. Note this happens during a Serviceguard rolling update configuration. Previous releases might be enhanced to recognize the new MWC format; contact your Hewlett-Packard support representative for up-to-the-moment information, or consult the Hewlett-Packard IT Resource Center site: <http://itrc.hp.com> (Americas and Asia Pacific) or <http://europe.itrc.hp.com> (Europe) for LVM patches.

NOTE: The new format of MWC introduced in HP-UX 11i v3 is supported on HP-UX 11i v2 with the patch PHKL_36244 or any superseding patch. With the patch PHKL_36244 or any superseding patch installed on HP-UX 11i v2, the new format of MWC is recognized and used to perform the recovery.

LVM Support for Large I/O Sizes

LVM now supports larger I/O sizes (up to the extent size). As I/O requests do not need to be split into smaller chunks, the I/O throughput is improved by issuing reads and writes of larger sizes through LVM.

Note that the actual I/O size might be limited by the interface card.

For more information, see the "HP-UX System Administrator's Guide: Logical Volume Management" available at <http://docs.hp.com/en/oshpux11iv3.html#LVM%20Volume%20Manager>

Also see the [The Next Generation Mass Storage Stack](#) whitepaper.

HP recommends you see the whitepaper for information on benefits and capabilities introduced in the New Mass Storage Stack in HP-UX 11i v3.

Parallel Resynchronization of Stale Extents

With September 2007 release of HP-UX 11i v3, LVM provides new command line options to the *lvextend*, *lvmerge*, and *lvsync* commands that control the synchronization of stale extents.

A new option, *-s*, is introduced in the *lvextend* and *lvmerge* commands, which suppresses the automatic synchronization of stale extents. This synchronization can be done by using the *lvsync* command. A new option, *-T*, is introduced in the *lvsync* command, which resynchronizes the specified logical volumes in parallel rather than serially. This reduces the time taken to synchronize the stale extents.

For more information, see *lvextend(1M)*, *lvmerge(1M)* and *lvsync(1M)*.

Parallel Open of Physical Volumes

In the September 2007 release of HP-UX 11i v3, a new command line option *-T* is introduced in the *vgchange* command that controls the way LVM activates a volume group. Normally, when a volume group is activated, all the physical volumes of the volume group are attached sequentially. Activating the volume group using *-T* option of the *vgchange* command attaches all the physical volumes in parallel. This reduces the time taken to activate multiple volume groups with large number of physical volumes.

For more information, see *vgchange(1M)*.

Increased Limits

Maximum Logical Volume Size Increased up to 16TB

LVM in HP-UX 11i v3 release supports logical volume of sizes up to 16 terabyte (TB) (versus 2TB in earlier releases).

In LVM, logical volume sizes must be a multiple of the extent size. Extent size is expressed in units of megabytes (MB) in the range of 1 to 256 and number of extents per logical volume is a decimal value in the range of 1 to 65535. Thus a logical volume with 65535 extents and 256MB extent size has the maximum supported size (16TB – 256MB).

The following are the advantages of 16TB logical volumes:

- These logical volumes offer better scalability.
- These logical volumes can be mirrored, striped, or striped and mirrored.
- These logical volumes are supported in standalone, Shared LVM, and exclusive mode.

Existing logical volumes using 256MB extents that are 2TB or less can be extended up to 16TB.

Note: For the logical volumes to extend up to 16TB, physical volumes in volume group must have sufficient LVM configuration space on the disk. To increase the LVM configuration space, see the [LVM Volume Group Dynamic LUN expansion \(DLE\)/vgmodify](#) whitepaper.

Compatibility Note

Releases prior to HP-UX 11i v3 can only access data within the first 2TB of a logical volume. If a volume group with logical volume larger than 2TB is created on HP-UX 11i v3, its activation and use is not recommended on any previous HP-UX release. Though the volume group can be activated and used on earlier releases, the data beyond 2TB is inaccessible.

NOTE: Logical volume with size greater than 2TB and up to 16TB is supported on HP-UX 11i v2 with patches PHCO_36744, PHKL_36745, and PHCO_36746. With these three patches or superseding patches installed on a HP-UX 11i v2 system, a volume group with such a logical volume created on HP-UX 11i v3 can be successfully imported and activated.

Increased Limits for Version 2.x Volume Groups

With the March and September 2008 releases of HP-UX 11i v3, LVM supports two new versions of volume group, Version 2.0 and Version 2.1 respectively. These new versions support configuration of larger number volume groups, logical volumes, physical volumes, and several other volume group parameters. For more information on the supported limits of Version 1.0 and Version 2.x volume groups, see the [LVM Limits White Paper](#).

HP MirrorDisk/UX software is supported in a clustered environment with a maximum of two nodes configured for volume group version 1.0 and 2.0. For volume group versions 2.1, HP MirrorDisk/UX software is supported in a clustered environment with a maximum of sixteen nodes configured.

Usability Enhancements

Compact, Parsable Command Output

LVM display commands have been enhanced to produce a compact and easily parsable output. A new option *-F* is introduced in *pvdisk*, *vgdisk*, and *lvdisk* commands. The commands generate colon-separated fields as `key=value[,value...]:key=value[,value]`.

The *-F* option is for use in scripts. The resulting command output can be split across multiple lines. The output can include new keys or values in the future. The positioning of a given key and value pair is not guaranteed. If a key is deprecated, its associated value is set to `NAM` (`key=NAM`). HP recommends all user scripts depending on LVM display commands' output to be modified to use *-F* option.

Examples

- Given a logical volume name and field name as specified in the *lvdisk -F* output, the following script extracts the value for the given key:

```
# cat get_lvinfo

#!/bin/sh

LV_NAME=$1;
LVDISPLAY_F_OUTPUT_KEY=$2;

lvdisk -F $LV_NAME | /usr/bin/awk -F':' '{for(i=1;i<=NF;i++) print $i;}' | \
    awk /$LVDISPLAY_F_OUTPUT_KEY/ | cut -f2 -d "="

# get_lvinfo /dev/vgtest/lvol1 lv_size
40
```

- Given a volume group name, the following script lists all the physical volume belonging to the volume group

```
# cat get_pv_in_vg

#!/bin/sh

VG_NAME=$1;

vgdisk -Fv $VG_NAME | sed -n -e "s/^. *pv_name=\\([^\.:]*\\):.*$/\\1/p"

# get_pv_in_vg /dev/vgtest
/dev/dsk/c11t0d5
/dev/dsk/c7t0d1
```

- Perl Sample example: Given a physical volume name and field name as specified in the *pvdisk -F* output , the following script extracts the value for the given key:

```
# cat get_pvinfos

#!/usr/bin/perl

my $PV_NAME = $ARGV[0];
my $PVDISPLAY_KEY = $ARGV[1];
my %pvdisk_hash = ();
```

```

my $pvdisplay_out = `pvdisplay -F $PV_NAME 2> /dev/null`;

if($? != 0) {
    print("The PV $PV_NAME does not belong to any VG\n");
    print("Provide a PV which belong to a VG\n");
    exit (1);
}

my @pvdisplay_pairs = split (' :', $pvdisplay_out);

foreach my $key_value (@pvdisplay_pairs) { my
    @pair = ();
    @pair = split('=', $key_value);
    $pvdisplay_hash{$pair[0]} = $pair[1];
}

print "$pvdisplay_hash{$PVDISPLAY_KEY}\n";

# get_pvinfos /dev/dsk/c11t0d5 vg_name
/dev/vgtest

```

pvdisplay – Displays user data offset and checks if disk is under HP-UX LVM control

The *pvdisplay* command displays information about physical volumes in LVM. It is enhanced to list additional details for a physical volume and to report if the disk is owned by LVM.

The *pvdisplay* command has the following new options:

- d
For each physical volume, displays the offset to the start of the user data in 1024 byte blocks from the beginning of the physical volume, specifies if *pv_path* is a bootable physical volume, and displays the number of bad blocks that were relocated. These details are displayed in addition to other information. Starting with the March 2008 release of HP-UX 11i v3, the offset to the end of the user data in 1024 byte blocks is also displayed.
- l
Checks whether physical volume refers to a disk device under HP Logical Volume Manager (LVM) control. You can use this option on any disk in the system. For example, on a system with many disks, use this option to determine which disks are marked for LVM usage and which are not.
- u
If physical volume is an LVM disk, displays the start and end of the user data in 1 kilobyte blocks. This option is supported starting with the March 2008 release of HP-UX 11i v3. In the September 2008 release of HP-UX 11i v3, this option is enhanced to display whether the physical volume is bootable or not.

Starting with the September 2007 release of HP-UX 11i v3, the *pvdisplay* command also displays the proactive polling status of the physical volume. For more information, see the *pvdisplay(1M)* manual page.

Examples

```

# pvdisplay -d /dev/dsk/c22t6d2

--- Physical volumes ---
PV Name           /dev/dsk/c22t6d2
VG Name           /dev/vgname
PV Status         available

```

Allocatable	yes
VGDA	2
Cur LV	5
PE Size (Mbytes)	4
Total PE	1279
Free PE	779
Allocated PE	500
Stale PE	0
IO Timeout (Seconds)	default
Autoswitch	On
Data Start	1024
Data End	34638688
Boot Disk	no
Relocated Blocks	61
Proactive Polling	On

To check whether the physical volume is owned by LVM, do the following:

```
# pvdisplay -l /dev/dsk/c4t3d0
/dev/dsk/c4t3d0:LVM_Disk=yes
```

```
# pvdisplay -l /dev/disk/disk96
/dev/disk/disk96:LVM_Disk=no
```

To find out the data start and data end of a bootable physical volume owned by LVM, do the following:

```
# pvdisplay -l -u /dev/disk/disk34_p2
/dev/disk/disk34_p2:LVM_Disk=yes>Data_Start=2912>Data_End=34638688:Bootable=yes
```

vgscan - Scans Faster, Per Volume Group Scanning and Supports Persistent DSF

The *vgscan* command recovers the LVM configuration files, */etc/lvmtab* and */etc/lvmtab_p* (introduced in March 2008 release of HP-UX 11i v3). It scans all physical volumes for LVM volume groups and adds entries for volume groups that are missing from */etc/lvmtab* and */etc/lvmtab_p*. The *vgscan* command recovers volume group information by using LVM data structures in kernel memory, and by probing all devices, searching for LVM disks.

In HP-UX 11i v3, *vgscan* has the following new options:

-f <vgs>

With this option *vgscan* works on per volume group basis rather than scanning for all the volume groups. This option replaces any existing entries related to these volume groups in */etc/lvmtab* or */etc/lvmtab_p* with updated entries. If the specified volume groups are missing, *vgscan* adds them to */etc/lvmtab* or */etc/lvmtab_p*.

Note: *vgscan -f* does not search for additional volume groups other than the one specified with the command and does not report unconfigured volume groups.

-k

This option provides a mechanism to make *vgscan* faster. It skips the disk probe portion of the *vgscan*, and retrieves volume group information only from the LVM data structures in kernel memory.

Note: With this option, *vgscan* only adds activated volume groups to */etc/lvmtab* and */etc/lvmtab_p*. For deactivated volume groups, no information is added to */etc/lvmtab* or */etc/lvmtab_p*.

-N

Recover */etc/lvmtab* and */etc/lvmtab_p* files using persistent DSFs, with the exception of active volume groups configured with legacy DSFs. In this case, *vgscan* populates the */etc/lvmtab* and */etc/lvmtab_p* files using legacy DSFs.

-B

Recover */etc/lvmtab* and */etc/lvmtab_p* files using both persistent and legacy DSFs. You can use this option to migrate a volume group configured with legacy DSFs to use corresponding persistent DSFs.

With the above options, *vgscan* provides the following improvements:

- With the *-f* option, an incorrect volume group entry in */etc/lvmtab* or */etc/lvmtab_p* can be corrected without modifying the rest of */etc/lvmtab* or */etc/lvmtab_p*. Previously, this could only be done in a cumbersome way using *vgexport* and *vgimport* to recover *lvmtab* or *lvmtab_p* entries for the root volume group.
- With the *-f* option, a user can specify the ordering of volume groups in */etc/lvmtab* and */etc/lvmtab_p*, such as placing the boot volume group first.
- With the *-k* option, *vgscan* can recover activated volume groups in */etc/lvmtab* and */etc/lvmtab_p* quickly (in seconds), even if the system has a large number of LVM disks configured.

Examples

- For the volume group */dev/vg01*, overwrite the existing physical volume DSFs in */etc/lvmtab* with the physical volume DSFs found belonging to */dev/vg01* during a hardware probe of all devices.

```
# vgscan -f /dev/vg01
```

```
Physical Volume "/dev/dsk/c2t0d0" contains no LVM information
Couldn't stat physical volume "/dev/dsk/c0t0d0":
Invalid argument
Physical Volume "/dev/dsk/c13t0d5" is not part of a Volume Group
Physical Volume "/dev/dsk/c11t0d5" is not part of a Volume Group
Physical Volume "/dev/dsk/c20t0d6" is not part of a Volume Group
Physical Volume "/dev/dsk/c17t0d6" is not part of a Volume Group
Physical Volume "/dev/dsk/c17t1d0" is not part of a Volume Group
Physical Volume "/dev/dsk/c20t1d0" is not part of a Volume Group
Physical Volume "/dev/dsk/c17t13d3" is not part of a Volume Group
Physical Volume "/dev/dsk/c20t13d3" is not part of a Volume Group
Physical Volume "/dev/dsk/c17t13d6" contains no LVM information
Physical Volume "/dev/dsk/c20t13d6" contains no LVM information
```

```
*** LVMTAB has been updated successfully.
```

```
*** Do the following to resync the information on the disk. ***
```

```
#1. vgchange -a y
```

```
*** #2. lvmboot -R
```

- For the volume group */dev/vg01*, overwrite the existing physical volume DSFs in */etc/lvmtab* with the physical volume DSFs used in kernel memory. The volume group */dev/vg01* must be activated, or this command will fail.

```
# vgscan -k -f /dev/vg01
```

- For the volume group */dev/vg01*, overwrite the existing physical volume DSFs in */etc/lvmtab* with the physical volumes persistent DSFs found belonging to */dev/vg01* during a hardware probing of all devices.

```
# vgscan -N -f /dev/vg01
```

```
vgscan: Warning: couldn't query physical volume "/dev/dsk/c7t0d1": The
specified path does not correspond to physical volume attached to this
volume group
vgscan: Warning: couldn't query physical volume "/dev/dsk/c9t0d1": The
specified path does not correspond to physical volume attached to this
volume group
vgscan: Warning: couldn't query all of the physical volumes.
vgscan: The physical volume "/dev/disk/disk25_p2" is already recorded in the
"/etc/lvmtab" file.
vgscan: The physical volume "/dev/disk/disk30" is already recorded in the
"/etc/lvmtab" file.
vgscan: The physical volume "/dev/disk/disk31" is already recorded in the
"/etc/lvmtab" file.
Physical Volume "/dev/disk/disk26" contains no LVM information
```

```

Couldn't stat physical volume "/dev/disk/disk27":
Invalid argument
Physical Volume "/dev/disk/disk36" is not part of a Volume Group
Couldn't stat physical volume "/dev/disk/disk80":
Invalid argument
Couldn't stat physical volume "/dev/disk/disk81":
Invalid argument
Physical Volume "/dev/disk/disk82" is not part of a Volume Group
Physical Volume "/dev/disk/disk83" is not part of a Volume Group
Physical Volume "/dev/disk/disk93" is not part of a Volume Group
Physical Volume "/dev/disk/disk96" contains no LVM information
*** LVMTAB has been updated successfully.
*** Do the following to resync the information on the disk. ***
#1. vgchange -a y
*** #2. lvmboot -R

```

- Recreate the /etc/lvmtab and /etc/lvmtab_p files for volume groups activated since the last boot. For activated and deactivated volume groups, use both persistent and legacy DSFs. Report all physical volume persistent and legacy DSFs belonging to unconfigured volume groups.

```

# mv /etc/lvmtab /etc/lvmtab.BCK
# mv /etc/lvmtab_p /etc/lvmtab_p.BCK
# vgscan -B
Creating "/etc/lvmtab".
Physical Volume "/dev/dsk/c4t12d0" is not part of a Volume Group
Physical Volume "/dev/dsk/c5t12d0" is not part of a Volume Group

```

```

Following Physical Volumes belong to one Volume Group.
Unable to match these Physical Volumes to a Volume Group.
Use the vgimport command to complete the process.
/dev/disk/disk35_p2
/dev/dsk/c4t1d0s2
/dev/dsk/c5t1d0s2

```

```

Following Physical Volumes belong to one Volume Group.
Unable to match these Physical Volumes to a Volume Group.
Use the vgimport command to complete the process.
/dev/disk/disk38_p2
/dev/dsk/c4t0d0s2
/dev/dsk/c5t0d0s2

```

```

Following Physical Volumes belong to one Volume Group.
Unable to match these Physical Volumes to a Volume Group.
Use the vgimport command to complete the process.
/dev/disk/disk48
/dev/dsk/c4t5d0
/dev/dsk/c5t5d0

```

```

Following Physical Volumes belong to one Volume Group.
Unable to match these Physical Volumes to a Volume Group.
Use the vgimport command to complete the process.
/dev/disk/disk45
/dev/dsk/c4t14d0
/dev/dsk/c5t14d0

```

```

Following Physical Volumes belong to one Volume Group.
Unable to match these Physical Volumes to a Volume Group.
Use the vgimport command to complete the process.

```



```
/dev/disk/disk46
/dev/disk/disk47
/dev/dsk/c4t13d0
/dev/dsk/c5t13d0
/dev/dsk/c4t10d0
/dev/dsk/c5t10d0
```

The Volume Group /dev/vg01 was not matched with any Physical Volumes.

```
*** LVMTAB has been created successfully.
*** If PV links are configured in the system.
*** Do the following to resync information on disk.
*** #1.  vgchange -a y
*** #2.  lvlnboot -R
Creating "/etc/lvmtab_p".
Couldn't stat physical volume "/dev/disk/disk44":
Invalid argument
Couldn't stat physical volume "/dev/disk/disk49":
Invalid argument
Couldn't stat physical volume "/dev/disk/disk40":
Invalid argument
Couldn't stat physical volume "/dev/dsk/c5t4d0":
Invalid argument
Couldn't stat physical volume "/dev/dsk/c4t4d0":
Invalid argument
Couldn't stat physical volume "/dev/dsk/c0t0d0":
Invalid argument
```

Following Physical Volumes belong to one Volume Group.
Unable to match these Physical Volumes to a Volume Group.
Use the vgimport command to complete the process.
/dev/disk/disk33
/dev/dsk/c2t1d0

Following Physical Volumes belong to one Volume Group.
Unable to match these Physical Volumes to a Volume Group.
Use the vgimport command to complete the process.
/dev/disk/disk37
/dev/disk/disk36
/dev/dsk/c4t2d0
/dev/dsk/c5t2d0
/dev/dsk/c4t9d0
/dev/dsk/c5t9d0

The Volume Group /dev/vgl21 was not matched with any Physical Volumes.
The Volume Group /dev/vgl1 was not matched with any Physical Volumes.
The Volume Group /dev/vgL202 was not matched with any Physical Volumes.
The Volume Group /dev/vg21 was not matched with any Physical Volumes.
*** LVMTAB has been created successfully.

For more information, see the vgscan(1M) manual page.

vgcfgrestore – Enhanced Listing of Backup File Contents

The *vgcfgrestore* command restores the LVM configuration data from a default (*-n* option) or alternate (*-f* option) configuration backup file to a physical volume, or it displays the configuration backup file content (*-l* option).

In HP-UX 11i v3, the *vgcfgrestore -l* command provides the following additional information from the configuration backup file when invoked together with the *-v* option:

- For each physical volume, display the disk size in kilobytes

- For each physical volume, display the starting block number (KB) of the user data
- For each physical volume, display the PVkey (See the `lvdisplay(1M)` manual page for more information).
- For each volume group, display `max_pv`, `max_pe`, and `max_lv`.

With the March 2008 release of HP-UX 11i v3, the `-v` option display the volume group version also. For a Version 2.x volume group, it also displays the maximum size to which the volume group can grow.

Example

```
# vgcfgrestore -l -v -n vg00
Volume Group Configuration information in "/etc/lvmconf/vg00.conf"
VG Name /dev/vg00
---- Physical volumes : 1 ----
   PV          Type          Size (kb)          Start (kb)  PVkey
   disk95_p2  Bootable          34643968          2912         0

max_pv 16 max_pe 4238 max_lv 255 vg_version 1.0
```

For a Version 2.0 volume group:

```
# vgcfgrestore -l -v -n vg02
Volume Group Configuration information in "/etc/lvmconf/vg02.conf"
VG Name /dev/vg02
---- Physical volumes : 1 ----
   PV          Type          Size (kb)          Start (kb)  PVkey
   disk50     Non-Boot          71687369          1024         0

max_pv 511 max_pe 2097152 max_lv 511 vg_version 2.0 vg_size 20t
```

Error Management Technology (EMT)

Error Management Technology (EMT) provides an online searchable repository containing error descriptions, probable causes and recommended actions on systems running HP-UX. With HP-UX 11i v3, LVM supports EMT and LVM error messages are available in the EMT repository.

Long Host Name Support for Display Commands

Shared LVM (SLVM) allows a volume group to be shared across different nodes within a Serviceguard cluster. In this environment, the `vgdisplay` command displays the host names of the nodes participating in the HP Serviceguard cluster. One of the host names is represented as "Server" and all other nodes as "Client".

In HP-UX 11i v3 the `vgdisplay` command has been enhanced to support large host names of more than 32 characters, (as in prior releases) up to 255 characters.

Note that though LVM supports host name length up to 255 characters, the host name displayed depends on the host name length supported by HP Serviceguard.

For more information, see the [Node and Host Name Sizes on HP-UX](#) whitepaper. The whitepaper provides information on supported host name lengths.

Example

Display the hostname information (machine-server and machine-client)

```
# vgdisplay -v /dev/vgtest1
--- Volume groups ---
VG Name                /dev/vgtest1
VG Write Access        read/write
VG Status              available, shared, server
Max LV                 255
Cur LV                1
Open LV                1
Max PV                 16
Cur PV                1
Act PV                1
Max PE per PV         8750
VGDA                   2
PE Size (Mbytes)      8
Total PE               8749
Alloc PE               50
Free PE               8699
Total PVG              0
Total Spare PVs       0
Total Spare PVs in use 0
```

```
machine1-server           Server
machine2-client         Client
```

```
--- Logical volumes ---
LV Name                /dev/vgtest1/lvol1
LV Status              available/syncd
LV Size (Mbytes)      400
Current LE             50
Allocated PE           50
Used PV                1
```

```
--- Physical volumes ---
PV Name                /dev/dsk/c4t13d0
PV Status              available
Total PE               8749
Free PE               8693
Autoswitch             On
```

lvmadm Command

With the March 2008 release of HP-UX 11i v3, LVM provides a new command, *lvmadm*, which displays the volume group limits for specified volume group versions supported by LVM.

In the September 2008 release of HP-UX 11i v3, this command is enhanced to support a new option, *-l*, which displays the configuration in the */etc/lvmtab* and */etc/lvmtab_p* files for the specified volume group version. This serves as a replacement for running the command `strings /etc/lvmtab` or `strings /etc/lvmtab_p`.

Examples

To view the supported limits for all volume group versions supported by LVM:

```
# lvmadm -t
--- LVM Limits ---
VG Version             1.0
Max VG Size (Tbytes)  510
Max LV Size (Tbytes)  16
Max PV Size (Tbytes)  2
```

Max VGs	256
Max LVs	255
Max PVs	255
Max Mirrors	2
Max Stripes	255
Max Stripe Size (Kbytes)	32768
Max LXs per LV	65535
Max PXs per PV	65535
Max Extent Size (Mbytes)	256
VG Version	2.0
Max VG Size (Tbytes)	2048
Max LV Size (Tbytes)	256
Max PV Size (Tbytes)	16
Max VGs	512
Max LVs	511
Max PVs	511
Max Mirrors	5
Max Stripes	511
Max Stripe Size (Kbytes)	262144
Max LXs per LV	33554432
Max PXs per PV	16777216
Max Extent Size (Mbytes)	256
VG Version	2.1
Max VG Size (Tbytes)	2048
Max LV Size (Tbytes)	256
Max PV Size (Tbytes)	16
Max VGs	2048
Max LVs	2047
Max PVs	2048
Max Mirrors	5
Max Stripes	511
Max Stripe Size (Kbytes)	262144
Max LXs per LV	33554432
Max PXs per PV	16777216
Max Extent Size (Mbytes)	256

To display the Version 1.0 volume groups and the corresponding physical volumes as contained in the LVM configuration file:

```
# lvmadm -l 1.0
--- Version 1.0 volume groups ---
VG Name /dev/vg00
PV Name /dev/disk/disk34_p2
```

For more information, see the `lvmadm(1M)` manual page.

vgremove – Removal of Volume Group Device Files

With the March 2008 release of HP-UX 11i v3, LVM supports a new `-X` option, which removes the volume group device files in `/dev`.

For more information, see `vgemove(1M)`.

vgdisplay – Displays More Information

With the March 2008 release of HP-UX 11i v3, *vgdisplay* displays the volume group version, the maximum size to which the volume group can grow and the maximum number of physical extents.

With the September 2008 release of HP-UX 11 iv3, LVM supports a new *-V* option, which displays the volume group information for all volume groups of the specified version.

Examples

```
# vgdisplay vg00
--- Volume groups ---
VG Name                /dev/vg00
VG Write Access        read/write
VG Status               available
Max LV                 255
Cur LV                 4
Open LV                4
Max PV                 16
Cur PV                 1
Act PV                 1
Max PE per PV          4238
VGDA                   2
PE Size (Mbytes)       8
Total PE               4228
Alloc PE                2029
Free PE                 2199
Total PVG               0
Total Spare PVs        0
Total Spare PVs in use 0
VG Version            1.0
VG Max Size           542464m
VG Max Extents       67808
```

To display all Version 1.0 volume groups on the system:

```
# vgdisplay -V 1.0
--- Volume groups ---
VG Name                /dev/vg00
VG Write Access        read/write
VG Status               available
Max LV                 255
Cur LV                 4
Open LV                4
Max PV                 16
Cur PV                 1
Act PV                 1
Max PE per PV          4238
VGDA                   2
PE Size (Mbytes)       8
Total PE               4228
Alloc PE                2029
Free PE                 2199
Total PVG               0
Total Spare PVs        0
Total Spare PVs in use 0
VG Version              1.0
VG Max Size             542464m
VG Max Extents          67808

VG Name                /dev/vg01
VG Write Access        read/write
VG Status               available
```

Max LV	255
Cur LV	1
Open LV	1
Max PV	16
Cur PV	1
Act PV	1
Max PE per PV	17501
VGDA	2
PE Size (Mbytes)	4
Total PE	17501
Alloc PE	10
Free PE	17491
Total PVG	0
Total Spare PVs	0
Total Spare PVs in use	0
VG Version	1.0
VG Max Size	1120064m
VG Max Extents	280016

For more information, see the `vgdisplay(1M)` manual page.

Auto Creation of Volume Group File

Prior to the March 2008 release of HP-UX 11i v3, the user manually created the directory for the volume group and the device file named 'group' using the `mkdir` and `mknod` commands.

With the March 2008 release of HP-UX 11i v3, the `vgcreate` and `vgimport` commands support automatic creation of the directory of the volume group and the volume group control file `/dev/vgname/group` regardless of the volume group version. The commands retain support for manually created `/dev/vgname/group` file. If the file is not present, the command creates one.

For more information, see the `vgcreate(1M)` and `vgimport(1M)` manual pages.

NOTE: Using the automatic creation of volume group control file feature, it is possible to have a shared volume group with different minor numbers on different nodes of a cluster. When creating shared volume groups, HP recommends that you manually create the group file with a minor number that is free on all nodes of the cluster.

pvmove Enhancements

In the September 2008 release of HP-UX 11i v3, the `pvmove` command is enhanced to have several new features:

- Support for moving a range of physical extents.
- Support for moving extents from the end of a physical volume.
- Support for moving extents to a specific location on the destination physical volume.
- Support for moving the physical extents from striped logical volumes and striped mirrored logical volumes.
- Support for a new option, `-p`, to preview physical extent movement details *without* performing the move.

Examples

- To preview the movement of first physical extent from source physical volume `/dev/dsk/c5t7d0` to any physical volume in the volume group:

```
# pvmove -p /dev/dsk/c5t7d0:0
```

```
--- Preview of the Relocation Operation ---
```

```
Source PV          Source PE  Destination PV          Destination PE
```

/dev/dsk/c5t7d0 0 /dev/dsk/c5t7d0 4

- To move extents in the range 25 to 100 from source physical volume /dev/disk/disk10 to the desired destination physical volume /dev/disk/disk22:
pvmove /dev/disk/disk10:25-100 /dev/disk/disk22
- To move the last 5 extents from /dev/disk/disk10 to another physical volume:
pvmove -e 5 /dev/disk/disk10
- To move extents to a specific location on the destination physical volume /dev/disk/disk22, starting at physical extent 50 on /dev/disk/disk22:
pvmove /dev/disk/disk10 /dev/disk/disk22:50

Miscellaneous

Commands enhanced to prevent misconfigurations through alternate links

Starting with the February 2007 release of HP-UX 11i v3, LVM commands were enhanced to avoid overwriting LVM configuration data on physical volumes that already belong to a volume group and are recorded in the `/etc/lvmtab` or `/etc/lvmtab_p`.

In prior releases, `pvcreate` with `-f` (force) option on an alternate physical volume path that was not present in `/etc/lvmtab` succeeded even though the corresponding physical volume belonged to a volume group recorded in `/etc/lvmtab`. This behavior has been changed with the February 2007 release of HP-UX 11i v3 so that `pvcreate` fails in such situations. The behavior is the same for both legacy and persistent DSFs of the physical volume.

In addition to `pvcreate` the following commands have been enhanced similarly: `vgcreate`, `pvremove`, `vgimport`, and `vgchgid`.

Example

As an example, consider a multipathed physical volume whose DSFs are `/dev/dsk/c9t1d0` and `/dev/dsk/c6t1d0`. Let the volume group be configured with only `/dev/dsk/c9t1d0`.

`'pvcreate -f /dev/rdisk/c6t1d0'` succeeds on releases prior to HP-UX 11i v3. But on HP-UX 11i v3 `pvcreate` fails and prints the following:

```
"pvcreate: The physical volume "/dev/dsk/c9t1d0" is already present in the
"/etc/lvmtab" file and represents an alternate link to "/dev/dsk/c6t1d0"."
```

Mirror Disk Installation No Longer Requires a Reboot

With the HP-UX 11i v3 release, the Mirror Disk product installation is much easier. Unlike previous releases, Mirror Disk product installation does not require a reboot.

Glossary

Agile Addressing

The ability to address a LUN with the same device special file regardless of the physical location of the LUN or the number of paths leading to it. In other words, the device special file for a LUN remains the same even if the LUN is moved from one HBA to another, moved from one switch/hub port to another, presented via a different target port to the host, or configured with multiple hardware paths. Also referred to as **persistent LUN binding**.

Agile View

The representation of LUNs using lunpath hardware paths, LUN hardware paths, and persistent DSFs, introduced in HP-UX 11i v3.

DSF

Device Special File. A file associated with an I/O device. DSFs are read and written the same as ordinary files, but requests to read or write are sent to the associated device.

Hardware Path

A series of numbers representing the physical or virtualized location of a device. The path is a sequence of I/O addresses that share a hierarchical relationship. The address elements might not correspond to physical hardware addresses, and might represent only a “handle” to a device rather than a physical path to it.

Legacy View

The representation of legacy hardware paths and legacy DSFs, as in releases prior to HP-UX 11i v3.

Legacy Hardware Path

The representation of a hardware path as it exists in releases prior to HP-UX 11i v3. It is composed of a series of bus-nexus addresses separated by '/' leading to the HBA; beneath the HBA, additional address elements (such as domain, area, port, target, and LUN) are separated by '.'. The string "0/2/1/0.1.4.0.0.2.7" is an example of a legacy hardware path.

Legacy DSF

A DSF with the hardware path information such as SCSI bus, target, and LUN embedded in the file's minor name and file name, such as `/dev/dsk/c2t3d4`.

LUN

A SCSI logical unit. This refers to an end storage device such as a disk, tape, floppy, or CD. This is the logical unit itself and does not represent the path to the logical unit.

Lunpath

The physical hardware path leading to a SCSI logical unit. A SCSI LUN can have more than one lunpath.

Persistent DSF

A DSF conforming to the naming model introduced in HP-UX 11i v3 to support agile addressing. The device file name contains an instance number, such as `/dev/disk/disk#`, and the minor number has no hardware path information.

Metadata

The on-disk structures that LVM uses to manage a volume group. This space is not available for application data.

Volume group version

Based on the metadata format, LVM supports three types of volume groups. Each type of volume group is assigned a version to distinguish from another type, based on the metadata format maintained by LVM.

For More Information

To learn more about some of the LVM features, see the following documents on HP documentation website:

<http://docs.hp.com> (Use search with the given name of the whitepaper)
<http://www.docs.hp.com/en/oshpux11iv3#LVM%20Volume%20Manager>

- SLVM Single-Node Online Reconfiguration (SLVM SNOR)
- LVM Online Disk Replacement
- LVM Volume Group Dynamic LUN expansion (DLE)/vgmodify
- LVM Volume Group Quiesce/Resume
- LVM Version 2.0 Volume Groups in HP-UX 11i v3
- LVM Limits White Paper

To learn more about configuring LVM and migration of LVM volume group configuration from legacy to agile naming model, see the following documents on HP documentation website:

<http://docs.hp.com> (Use search with the given name of the whitepaper)
<http://www.docs.hp.com/en/oshpux11iv3#LVM%20Volume%20Manager>

- LVM Migration from Legacy to Agile Naming Model
- HP-UX System Administrator's Guide Logical Volume Management

To learn more about the agile view and the new mass storage stack, see the following document on HP documentation website: <http://docs.hp.com/en/netsys.html#Storage%20Area%20Management>

- The Next Generation Mass Storage Stack

To learn more about supported node and host name sizes on HP-UX, see the following document on HP documentation website: <http://www.docs.hp.com/en/oshpux11iv3#White%20Papers>

- Node and Host Name Sizes on HP-UX

Call to Action

HP welcomes your input. Please give us comments about this whitepaper, or suggestions through our technical documentation feedback website: <http://docs.hp.com/en/feedback.html>.

© 2008 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.