

LVM Volume Group Dynamic LUN expansion (DLE) / vgmodify



Abstract.....	2
Overview.....	3
What can vgmodify do for me?	3
Prerequisites	3
When would I want to use vgmodify?.....	3
Normal procedure for using vgmodify	3
Increasing the range of values vgmodify provides.....	4
Details.....	5
Changing disk type (boot/non-boot) and impact of boot disks.....	5
Dynamic LUN expansion (DLE) and LUN contraction	5
Physical extent renumbering (-n option) and pvmove enhancement.....	6
How the volume group parameter settings affect the metadata size.....	6
Configuration backup files, restore script and vgcfgrestore changes.....	6
Making better use of the alternate bad block pool space	8
Optimization of max_pv and max_pe values	8
Guidance on choosing the max_pv and max_pe values.....	8
Common usage examples.....	9
Increasing max number of physical volumes and accommodating larger disks – <i>without</i> PE renumbering... ..	9
Increasing max number of physical volumes and accommodating larger disks – <i>with</i> PE renumbering	12
Changing a bootable disk to non-bootable	17
Optimizing the metadata space.....	21
Handling dynamic LUN expansion and configurations where not all the disk space can be allocated	23
LVM vgmodify Messages	27
vgmodify(TM).....	27
For More Information	29
Call to action	29

Abstract

The new logical volume manager `vgmodify` command allows changing attributes that previously were fixed at creation time, for example increasing the maximum number of physical volumes or the maximum number of extents. It also includes the ability to handle the resizing of disks (Dynamic LUN Expansion - DLE) which is a feature commonly provided by disk arrays. This paper explains the benefits of `vgmodify` and how to use it.

It is available on HP-UX 11i v2 with PHCO_35524 and 11i v3.

Overview

What can `vgmodify` do for me?

The new `vgmodify` command provides the ability to alter some characteristics of an existing Volume Group (VG) or Physical Volume (PV) that were set via `vgcreate` or `pvcreate`. These modifiable attributes are the maximum number of PVs, the maximum number of Logical Volumes (LVs) and the maximum number of extents per PV allowed in the VG, as well as whether a PV should be bootable or not. The `vgmodify` command handles the expansion (DLE) and contraction of a LUN.

Before the introduction of `vgmodify` all these operations required the data in the volume group to be backed up and the volume group destroyed, or physical volume removed, before rebuilding and reloading the data.

Prerequisites

- The volume group must be deactivated when changes are being made (for the root volume group this means booting into maintenance mode).
- The volume group must be cluster unaware (`vgchange -c n`) before changes are made. After `vgmodify` has altered the volume group the user can then make the volume group cluster aware (`vgchange -c y`).
- All physical volumes belonging to volume group must be accessible.
- There cannot be any entries in the bad block directory of any physical volume in the volume group. The `vgmodify` command will report any disks containing bad blocks and exit without making any changes. Use `pvdisplay -d` to display the count of bad blocks and verify this is the cause of the failure. To overcome this problem the disk can be replaced and the configuration restored via `vgcfgrestore`.

When would I want to use `vgmodify`?

- To add more physical volumes than the max PV setting currently allows.
- To take account of a size change to a physical volume since it was initialized via `pvcreate`.
- To make full use of the space available on a disk which is larger than the current maximum number of physical extents per physical volume allows.
- To change a disk type (boot to non-boot or vice versa).
- To change the maximum number of logical volumes in the volume group.
- To make full use of the LVM metadata space.

Normal procedure for using `vgmodify`

1. Not all volume group selections are available so determine what available settings best meet your needs. Note: The `vgmodify -t` operation can be run against an activated or deactivated volume group.

```
# vgmodify -t -v vg02           # Table without physical extent renumbering  
                                (see the section on physical extent renumbering for more details)  
# vgmodify -t -v -n vg02       # Table with physical extent renumbering
```

2. Review the output and select the new values. The user is responsible for freeing up the first physical extent on any physical volume as required by using `pvmove`, `lvreduce` or `lvremove`.

3. Check that these can be applied. This step is particularly recommended when the volume group is currently active and in use. This step can be performed on an activated or deactivated volume group.

```
# vgmodyfy -r -v -e 2000 -p 150 vg02          # Note -r, review mode.
```

4. During a maintenance window, deactivate the volume group (for the root VG boot the volume group into maintenance mode), make it cluster unaware (if necessary) and run `vgmodyfy` to apply the new settings.

Non-root volume group.

```
# vgchange -a n vg02          # Deactivate the volume group.
# vgmodyfy -v -e 2000 -p 150 vg02  # Same options and arguments as in
                                   # the previous step but without -r.
# vgchange -a y -s vg02      # Activate the volume group.
```

Root volume group (boot into maintenance mode).

```
# cd / ; shutdown -r now
```

Interact with boot process

```
ISL> hpux -lm
```

```
# vgmodyfy -v -e 2000 -p 150 vg02  # Same options and arguments as in
                                   # the previous step but without -r.
# vgchange -a y -s vg02 # Verify the volume group activates with the new settings
# vgchange -a n vg02    # Deactivate volume group
# reboot
```

Increasing the range of values vgmodyfy provides

The following method describes how one or more of the physical volumes can be moved into another volume group. Having carried out this task new disks can be added or `vgmodyfy` used to increase the maximum number of physical extents per physical volume. This is only possible where the volume group (VG) has one group of logical volumes (LVs) using one set of physical volumes (PVs) and the remaining logical volumes using a different set of physical volumes (it may be possible to use `pvmove` to re-organize the layout if it does not already conform).

E.g. VG has LVs 1, 2 and 3 on PVs 0 and 1. LVs 4, 5 and 6 on PVs 2, 3, 4 and 5.

```
# vgchange -a n VG
# vgexport -m MAP VG
# vgchgid PV2 PV3 PV4 PV5
# mkdir /dev/VG; mknod /dev/VG/group c 64 0xnn0000
                                   # Where nn is the original volume group number.
# mkdir /dev/new_VG; mknod /dev/new_VG/group c 64 0xmm0000
                                   # Where mm is the new volume group number.
# vgimport -m MAP /dev/VG PV0 PV1
# vgimport -m MAP /dev/new_VG PV2 PV3 PV4 PV5
# vgchange -a y -q n VG; vgchange -a y new_VG
# lvremove -f VG/LV4 VG/LV5 VG/LV6 new_VG/LV1 new_VG/LV2 new_VG/LV3
# vgreduce -f VG; vgreduce -f new_VG
# vgcfgbackup VG; vgcfgbackup new_VG
```

On completion, VG holds three LVs (1, 2 & 3) with PVs 0 & 1. Whilst new_VG holds three LVs (4, 5 & 6) across PVs 2, 3, 4 & 5.

Remember to change any name references to the new volume group (e.g. If the old LV4 is recorded in `/etc/fstab` it must be changed to `/dev/new_VG/LV4`).

Details

The `vgmodify` command is invoked in one of three modes, table, validate or change:

- Table mode. A table (-t) of possible settings are displayed.
- Review mode. A review (-r) is performed that the new settings, based upon the other arguments, can be successfully applied.
- Change mode. New settings are applied to the volume group.

Normally these would be performed in this sequence. I.e. table first, validate second and finally change mode.

If the volume group is deactivated, then it is activated by `vgmodify` and `vgcfgbackup` called. The `vgmodify` command rejects starting on an active volume group if it is invoked in change mode (this is necessary as `vgmodify` must be in full control of the volume group, it must be able to choose the options used to activate it and have the ability to deactivate as necessary). The configuration backup file (created by `vgcfgbackup`) is used to determine the volume group layout. The current size of each physical volume is obtained by interrogating each one directly.

Changing the volume group parameters affects the size of the LVM metadata. An area at the front of each physical volume is reserved to hold the LVM metadata. The `vgmodify` command can expand the LVM metadata if space before the first physical extent is available. It is also possible for `vgmodify` to utilize (where the `-n` option is used) the first physical extent, if free, for LVM metadata.

Where the options require it (i.e. change mode and changes are required) `vgmodify` builds a new configuration backup file based upon the new layout. The `vgmodify` command then invokes `vgcfgrestore` to apply this configuration to all of the physical volumes.

Changing disk type (boot/non-boot) and impact of boot disks

- If a non-boot volume group contains bootable physical volumes then the volume group parameters available for selection are constrained by the boot disk. Therefore by changing all boot disks in a non-boot volume group to non-boot, a wider choice of settings are available. Hence run `vgmodify` with '`-B n`' and a trailing list of all bootable physical volumes with all the `vgmodify` commands (including the `-t` option) where this is the situation. Disk type can be identified by running `vgcfgrestore -l -n <vg name>` or `pvdisplay -d`.
- If a physical volume is incorrectly converted from bootable to non-bootable then `vgmodify` can be used to reinstate the type. But it should be noted that for a physical volume to be converted to a boot disk then either all its extents must be free or the first physical extent must start at block 2912 (as a user you cannot directly control the start block for the first physical extent, conversion to boot from non-boot will normally only be possible if `vgmodify` was previously used to convert the same disk from boot to non-boot). Also the size of the LVM metadata structure VGRA (Volume Group Reserve Area) can be no greater than 768Kb. To complete the process run `mkboot` and `lvlnboot` to rebuild the BDRA metadata.

Dynamic LUN expansion (DLE) and LUN contraction

When a LUN is dynamically grown, `vgmodify` should be used to allow LVM to access this new space. Once the volume group has been adjusted by `vgmodify` the new space can be allocated using the normal LVM method by `lvextend` or `lvcreate`.

- Every time `vgmodify` is run it automatically obtains the current disk size and compares this with the size known to LVM. No additional options are required to handle a DLE event.

- If `pvcreate` was originally run with `-s` and a lower disk size value `vgmodify` adjusts the setting to match its actual size.
- DLE may require that the `max_pe` value is increased so that all the space can be allocated to physical extents. This can be achieved by including the `-e` option and an increased `max_pe` setting.
- Before contracting a physical volume make certain the space being reduced is not associated with allocated extents.
- The `vgmodify` command performs a physical volume size reduction only if the space being removed is held in extents that are free. If the extents are in use `vgmodify` reports this condition and exits without making changes. To overcome this problem the physical volume should be increased back to its original size (where possible) and then the extents in use made free before again reducing the size of the disk. Where it is not possible to increase the size of the physical volume, the extents associated with the device should be made free via `lvreduce -k` or `lvremove`.

Physical extent renumbering (-n option) and pvmove enhancement

A side effect of changing the volume group parameters is that the LVM metadata (VGRA) may change size. The maximum size of the VGRA is limited by the physical extent size and it is replicated on all physical volumes in the volume group. To allow the VGRA to grow to its maximum size it may be necessary to make the first physical extent on each physical volume in the volume group free. This must be accomplished outside of `vgmodify` by using `pvmove`, `lvreduce` and/or `lvremove`.

An enhancement to the `pvmove` command allows the user to move the contents of physical extent zero thereby making it free.

A subsequent `vgmodify` command can reallocate an unused physical extent zero for metadata use. A consequence of this is that physical extents will be renumbered lower (the old physical extent one becoming new physical extent zero). Logical extent numbering remains unchanged.

If the specified volume group parameters are sufficiently smaller than the existing settings it is possible that the new VGRA is small enough to allow metadata space to be returned to physical extent space. When this occurs physical extents will be renumbered higher (space that was partially metadata now holding the new physical extent zero (automatically marked free) and the old physical extent zero becoming new physical extent one). Logical extent numbering remains unchanged.

Where a configuration permits both higher and lower renumbering (for example all physical extents free on all physical volumes in the volume group) `vgmodify` favors renumbering higher.

How the volume group parameter settings affect the metadata size

The choice of volume group settings is affected by the LVM metadata size. Whilst the `max_lv` setting has only a small impact on the VGRA size (by default `vgcreate` has already set `max_lv` to the maximum possible value) the `max_pv` and `max_pe` settings are a major factor. As a general rule by reducing either the `max_pe` or `max_pv` values an increase in the other (`max_pe` or `max_pv`) can be accommodated. As a helpful approximation halving one (`max_pe` or `max_pv`) allows the doubling of the other. For example, if you halve `max_pe` you will be able to approximately double `max_pv`.

Configuration backup files, restore script and vgcfgrestore changes

The `vgmodify` command uses LVM configuration backup files in its operations. These backup files are used as a source of information about the volume group, and if needed can be used for recovery operations. There are a few items to keep in mind.

- Each time `vgmodify` is used a new set of configuration backup file are created. This is done each time to ensure it is using the most up to date configuration details.
- The configuration files are stored in the `/etc/lvmconf` directory.
- Unless being run in review (`-r`) or table (`-t`) mode, `vgmodify` uses `vgcfgrestore` to apply the configuration to all the physical volumes. Should the operation not be fully completed, for example it is interrupted by a system crash, a manual restoration can be performed. To simplify a recovery `vgmodify` creates a `<vg_name>_restore` script under `/etc/lvmconf`. This shell script is not used by `vgmodify` and is only provided for manual restoration to all physical volumes, taking into account any alternate paths.

With the introduction of `vgmodify` the configuration files for a specific volume group may have differing parameters. The `vgcfgrestore` command has been enhanced and now offers a `-v` option, which is only applicable with `-l`, to additionally report from the configuration backup file:

- The size of each physical volume.
- The start block for physical extent zero of each physical volume.
- The PVkey (see the `-k` option on `lvsdisplay(1M)`) of each physical volume (paths with the same PVkey are links (PVlinks) to the same physical volume).
- The `max_pv` (maximum number of physical volumes in the volume group)
- The `max_pe` (maximum number of physical extents per physical volume)
- The `max_lv` (maximum number of logical volumes in the volume group) for the volume group.

Once `vgmodify` has performed a modification all old configuration backup files should be removed. However if some do remain, and a restore is to be performed, use extreme care in using older configuration files. A `vgcfgrestore -v1` should be used on the file and verify the values match the current configuration. Compare the output of `vgcfgrestore -v1` on the two files. If the `max_pv`, `max_pe` and/or `max_lv` values differ then use the output of `vgdisplay` to validate which set is current. If the size of a PV is different then use `diskinfo` on the raw device to check the current size. Manually remove old `lvmconf` files that hold out of date information.

Example output:

- **With -v**

```
# vgcfgrestore -v1 -n vg01
Volume Group Configuration information in "/etc/lvmconf/vg01.conf"
VG Name /dev/vg01
---- Physical volumes : 2 ----
   PV          Type          Size (kb)          Start (kb)  PVkey
   c3t1d0      Non-Boot      1000000            1024         0
   c2t1d0      Non-Boot      1000000            1024         0

max_pv 16  max_pe 1016 max_lv 255
```

- **Without -v**

```
# vgcfgrestore -l -n vg01
Volume Group Configuration information in "/etc/lvmconf/vg01.conf"
VG Name /dev/vg01
---- Physical volumes : 2 ----
   /dev/rdsk/c3t1d0 (Non-bootable)
   /dev/rdsk/c2t1d0 (Non-bootable)
```

Making better use of the alternate bad block pool space

Patches on 11i v1 and v2 have resulted in the bad block directory only being used for entries of pre-existing bad blocks as modern disk devices can perform their own handling of media errors. If the bad block directory is empty, `vgmodify` tries to return the space reserved for the alternate bad block pool to user data. In some cases the alternate bad block pool on a physical volume is greater than the extent size. When this is the case, and the `max_pe` is adequate, `vgmodify` increases the number of physical extents for the physical volume. When this happens, the number of free and total number of physical extents on the physical volume increases.

Optimization of `max_pv` and `max_pe` values

Or, why doesn't `-t` show all possible settings and what does the `-o` option do?

The values displayed in the `-t` output are not a complete set of all possible choices but instead are the optimal ones. For instance if the output shows two adjacent lines:

<code>max_pv(-p)</code>	<code>max_pe(-e)</code>	Disk size (Mb)
35	2556	10224
39	2300	9200

A `max_pv` of 36, 37 or 38 can still be applied to the volume group but the highest `max_pe` for all of these is the same as for 39, i.e. 2300. Similarly a `max_pe` between 2557 and 2299 can be used but the highest `max_pv` is 39. In actual fact with a `max_pv` of 39 the `max_pe` can range from the highest physical extent in use on any physical volume in the volume group to 2300 and with a `max_pe` of 2300 the `max_pv` can range from the number of physical volumes currently part of the volume group to 39. So the `-t` output shows the optimal values not the complete list. Or to put it another way the most sensible set of values to use.

The `-o` option instructs `vgmodify` to use the highest optimal `max_pv` and `max_pe` values. This option would most commonly be used on volume groups which have not been modified by `vgmodify` and where the user has no specific need to change any value. It is asking `vgmodify` to make best use of the metadata space.

Guidance on choosing the `max_pv` and `max_pe` values

When adjusting sizes of an existing or creating a new volume group, picking `max_pv` and `max_pe` has some side effects on the overall volume group configuration. It is best to provision for growth as best as possible. In general, the larger the extent size (this is the one parameter `vgmodify` cannot alter) the larger the volume group can grow. The following are some guidelines.

- From the table mode (the `-t` option) you can see how `max_pv`, `max_pe` and disk size can be tuned for your specific needs or growth plans.
- If you have plans to increase the number of physical volumes in the volume group, use the disk size and `max_pv` columns from the `'-t'` table to determine the `max_pv` setting. You should review the Logical Volume Management "Configuring for performance section" in the HP-UX Administrator's Guide for guidance on the number of volume groups. In general try to keep the number of physical volumes to a minimum to improve performance.
- If you have plans to increase the largest size of disk in the volume group, use the `max_pv` and `max_pe` columns from the `'-t'` table to determine the size that best meets your needs.
- Choose a setting from the `'-t'` table which gives an adequate number of physical volumes of the largest size required.
- Physical extent renumbering will enable more flexibility for picking `max_pv` and `max_pe`. If the `'-t'` table does not provide suitable values then check the `'-tn'` table.

Common usage examples

It should be noted that the examples assume the volume group is already activated, but this is not a requirement, as `vgmodify` will activate the volume group if needed (if `vgmodify` does activate the volume group then it will deactivate before completion).

Increasing max number of physical volumes and accommodating larger disks – *without* PE renumbering

Using `vgmodify -t`, select the entry from the tables which meets the requirement for the maximum disk size with an adequate number of physical volumes. If you cannot achieve the number of physical volumes you require, use the `'-tn'` option on `vgmodify`. If a setting from the `-tn` table is to be used and this requires physical extent renumbering lower then run `vgmodify -tnv` and check that no disks need the first physical extent to be made free. If they do, then free up this extent on each disk using `pvmove`, `lvreduce` and/or `lvremove`. Run `vgmodify -r -e max_pe -p max_pv <vgname>`, where `max_pe` and `max_pv` are the chosen values. Finally, when a maintenance window is available, deactivate the volume group and run `vgmodify` with the same options as before but without `-r`.

1. In this example we want to place a disk that is 20Gb in size and increase the number of physical volumes to at least 20.

```
# vgmodify -tv vg02
```

```
Volume Group configuration for /dev/vg02 has been saved in
/etc/lvmconf/vg02.conf
```

```
Current Volume Group settings:
```

Max LV	255
Max PV	16
Max PE per PV	1023
PE Size (Mbytes)	4
VGRA Size (Kbytes)	208

```
VGRA space (Kbytes) on Physical Volumes with extents in use:
```

PV	current	-n
/dev/rdisk/c0t0d0	896	4096
/dev/rdisk/c1t0d0	896	4096
Summary	896	4096

```
Volume Group optimized settings (no PEs renumbered):
```

max_pv(-p)	max_pe(-e)	Disk size (Mb)
2	53756	215025
* Entries deleted for brevity *		
16	6652	26609
17	6140	24561
18	5884	23537
19	5628	22513
20	5372	21489
21	5116	20465
22	4860	19441
23	4604	18417
* Entries deleted for brevity *		
70	1532	6129
84	1276	5105

Table mode displays the current settings, including the current VGRA size. The `-v` argument shows the amount of space for the VGRA on each of the physical volumes. The summary

shows that the VGRA can be increased to 896Kb without using the `-n` argument or 4096Kb (the PE size) with `-n`. The list of settings is based upon a VGRA no greater than 896Kb in size.

2. There are two optimal settings that allow an 20Gb disk to be part of the volume group with at least 20 physical volumes. We'll select a `max_pv` of 21.

```
# vgmodify -r -v -p 21 -e 5116 vg02
```

```
Volume Group configuration for /dev/vg02 has been saved in  
/etc/lvmconf/vg02.conf
```

```
Current Volume Group settings:
```

```
Max LV      255  
Max PV      16  
Max PE per PV 1023  
PE Size (Mbytes) 4  
VGRA Size (Kbytes) 208
```

```
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      21  
Max PE per PV 5116  
PE Size (Mbytes) 4  
VGRA Size (Kbytes) 896
```

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

The review mode (`-r`) displays the current and new settings for comparison. The program completes before making any changes.

3. Deactivate the volume group in preparation for the change.

```
# vgchange -a n vg02
```

```
Volume group "vg02" has been successfully changed.
```

4. Apply the changes using the same `vgmodify` arguments but without `-r`.

```
# vgmodify -v -p 21 -e 5116 vg02
```

```
Performing "vgchange -a y -l -p -s vg02" to collect data  
Activated volume group  
Volume group "vg02" has been successfully activated.  
Volume Group configuration for /dev/vg02 has been saved in  
/etc/lvmconf/vg02.conf  
Deactivating Volume Group "vg02"  
Volume group "vg02" has been successfully deactivated.
```

```
Current Volume Group settings:
```

```
Max LV      255  
Max PV      16  
Max PE per PV 1023  
PE Size (Mbytes) 4  
VGRA Size (Kbytes) 208
```

```
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      21  
Max PE per PV 5116  
PE Size (Mbytes) 4
```

```

New Volume Group configuration for "vg02" has been saved in
"/etc/lvmconf/vg02.conf"
Old Volume Group configuration for "vg02" has been saved in
"/etc/lvmconf/vg02.conf.old"
Starting the modification by writing to all Physical Volumes
Applying the configuration to all Physical Volumes from
"/etc/lvmconf/vg02.conf"
Volume Group configuration has been restored to /dev/rdisk/c0t0d0
Volume Group configuration has been restored to /dev/rdisk/c1t0d0
Completed the modification process.
New Volume Group configuration for "vg02" has been saved in
"/etc/lvmconf/vg02.conf.old"
Volume group "vg02" has been successfully changed.

```

Almost all the initial output is identical to that for `-r`. The underlined text is only applicable when a change is being made. Note the program reports whether the `VG.conf` and/or `VG.conf.old` contains the new or old layout at various stages (useful should it be interrupted).

5. Activate the volume group.

```

# vgchange -a y -s vg02
Volume group "vg02" has been successfully changed.

```

6. Check the values are now correct.

```

# vgdisplay vg02
--- Volume groups ---
VG Name                /dev/vg02
VG Write Access        read/write
VG Status              available
Max LV                 255
Cur LV                1
Open LV                1
Max PV                 21
Cur PV                2
Act PV                 2
Max PE per PV         5116
VGDA                   4
PE Size (Mbytes)      4
Total PE               2046
Alloc PE               1800
Free PE                246
Total PVG              0
Total Spare PVs       0
Total Spare PVs in use 0

```

Note the Max PE per PV and Max PV settings match the requested setting.

Increasing max number of physical volumes and accommodating larger disks – *with* PE renumbering

Using `vgmodify -t`, select the entry from the tables which meets the requirement for the maximum disk size with an adequate number of physical volumes. If you cannot achieve the number of physical volumes you require, use the `'-tn'` option on `vgmodify`. If a setting from the `-tn` table is to be used and this requires physical extent renumbering lower then run `vgmodify -tnv` and check that no disks need the first physical extent to be made free. If they do, then free up this extent on each disk using `pvmove`, `lvreduce` and/or `lvremove`. Run `vgmodify -r -e max_pe -p max_pv <vgname>`, where `max_pe` and `max_pv` are the chosen values. Finally, when a maintenance window is available, deactivate the volume group and run `vgmodify` with the same options as before but without `-r`.

1. In this example we want to place a disk that is 80Gb in size and increase the number of physical volumes to at least 20.

```
# vgmodify -tv vg01
```

```
Volume Group configuration for /dev/vg01 has been saved in  
/etc/lvmconf/vg01.conf
```

```
Current Volume Group settings:
```

Max LV	255
Max PV	16
Max PE per PV	17366
PE Size (Mbytes)	4
VGRA Size (Kbytes)	2272

```
VGRA space (Kbytes) on Physical Volumes with extents in use:
```

PV	current	-n
/dev/rdisk/c11t0d0	2304	4096
/dev/rdisk/c11t1d0	2304	4096
/dev/rdisk/c11t2d0	2304	4096
/dev/rdisk/c11t3d0	2304	4096
Summary	2304	4096

```
Volume Group optimized settings (no PEs renumbered):
```

max_pv(-p)	max_pe(-e)	Disk size (Mb)
* Entries deleted for brevity		
13	21500	86002
14	19964	79858
15	18684	74738
16	17404	69618
17	16380	65522
18	15612	62450
19	14844	59378
20	14076	56306
21	13308	53234
* Entries deleted for brevity		

Note without `-n` that we can only place 13 disks of the required size or greater into the volume group. Also that the summary shows that with `-n` we'll be able to handle a larger VGRA so using `-n` should be of benefit.

```
# vgmodify -tnv vg01
```

```
Volume Group configuration for /dev/vg01 has been saved in
/etc/lvmconf/vg01.conf
```

```
Current Volume Group settings:
```

Max LV	255
Max PV	16
Max PE per PV	17366
PE Size (Mbytes)	4
VGRA Size (Kbytes)	2272

```
VGRA space (Kbytes) on Physical Volumes with extents in use:
```

PV	current	-n
/dev/rdisk/c11t0d0	2304	4096
/dev/rdisk/c11t1d0	2304	4096
/dev/rdisk/c11t2d0	2304	4096
/dev/rdisk/c11t3d0	2304	4096
Summary	2304	4096

```
Physical Extent zero is not free on all PVs. You will not achieve
these
values until the first extent is made free (see pvmove(1M)) on all
the
following disks:
/dev/rdisk/c11t0d0
/dev/rdisk/c11t1d0
/dev/rdisk/c11t2d0
/dev/rdisk/c11t3d0
```

```
Volume Group optimized settings (PEs renumbered lower):
```

max_pv(-p)	max_pe(-e)	Disk size (Mb)
* Entries deleted for brevity		
20	14076	56306
21	23804	95220
22	22780	91124
23	21756	87028
24	20988	83956
25	19964	79860
* Entries deleted for brevity		

With -n we can place up to 24 disks of 35Gb in size but we must free up the first extent on four disks. The desire was for at least 20 disks so any entries between the 20 and 24 range will suffice. We'll use 24.

- Free up the first extent on all four disks – note: keeping the extent on the same disk.

```
# pvmove /dev/dsk/c11t0d0:0 /dev/dsk/c11t0d0
```

```
Transferring logical extents of logical volume "/dev/vg01/lvol1"...
Physical volume "/dev/dsk/c11t0d0" has been successfully moved.
Volume Group configuration for /dev/vg01 has been saved in
/etc/lvmconf/vg01.conf
```

```
# pvmove /dev/dsk/c11t1d0:0 /dev/dsk/c11t1d0
```

```
Transferring logical extents of logical volume "/dev/vg01/lvol1"...
Physical volume "/dev/dsk/c11t1d0" has been successfully moved.
Volume Group configuration for /dev/vg01 has been saved in
/etc/lvmconf/vg01.conf
```

```
# pvmove /dev/dsk/c11t2d0:0 /dev/dsk/c11t2d0
Transferring logical extents of logical volume "/dev/vg01/lvol1"...
Physical volume "/dev/dsk/c11t2d0" has been successfully moved.
Volume Group configuration for /dev/vg01 has been saved in
/etc/lvmconf/vg01.conf
```

```
# pvmove /dev/dsk/c11t3d0:0 /dev/dsk/c11t3d0
Transferring logical extents of logical volume "/dev/vg01/lvol1"...
Physical volume "/dev/dsk/c11t3d0" has been successfully moved.
Volume Group configuration for /dev/vg01 has been saved in
/etc/lvmconf/vg01.conf
```

3. Review mode.

```
# vgmodify -r -v -n -p 24 -e 20988 vg01
Volume Group configuration for /dev/vg01 has been saved in
/etc/lvmconf/vg01.conf
```

Current Volume Group settings:

Max LV	255
Max PV	16
Max PE per PV	17366
PE Size (Mbytes)	4
VGRA Size (Kbytes)	2272

The current and new Volume Group parameters differ.
"/dev/rdisk/c11t0d0" new configuration requires PEs are renumbered lower
"/dev/rdisk/c11t1d0" new configuration requires PEs are renumbered lower
"/dev/rdisk/c11t2d0" new configuration requires PEs are renumbered lower
"/dev/rdisk/c11t3d0" new configuration requires PEs are renumbered lower
An update to the Volume Group IS required

New Volume Group settings:

Max LV	255
Max PV	24
Max PE per PV	20988
PE Size (Mbytes)	4
VGRA Size (Kbytes)	4096

Review complete. Volume group not modified

Note that all four disks that the first extent was moved on are reported as having PEs renumbered lower.

4. Deactivate the volume in readiness for the modification.

```
# vgchange -a n vg01
Volume group "vg01" has been successfully changed.
```

5. Change mode.

```
# vgmodify -v -n -p 24 -e 20988 vg01
Performing "vgchange -a y -l -p -s vg01" to collect data
Activated volume group
Volume group "vg01" has been successfully activated.
Volume Group configuration for /dev/vg01 has been saved in
/etc/lvmconf/vg01.conf
Deactivating Volume Group "vg01"
Volume group "vg01" has been successfully deactivated.
```

Current Volume Group settings:

Max LV	255
Max PV	16
Max PE per PV	17366
PE Size (Mbytes)	4
VGRA Size (Kbytes)	2272

The current and new Volume Group parameters differ.

"/dev/rdisk/c11t0d0" new configuration requires PEs are renumbered lower

"/dev/rdisk/c11t1d0" new configuration requires PEs are renumbered lower

"/dev/rdisk/c11t2d0" new configuration requires PEs are renumbered lower

"/dev/rdisk/c11t3d0" new configuration requires PEs are renumbered lower

An update to the Volume Group IS required

New Volume Group settings:

Max LV	255
Max PV	24
Max PE per PV	20988
PE Size (Mbytes)	4
VGRA Size (Kbytes)	4096

New Volume Group configuration for "vg01" has been saved in
"/etc/lvmconf/vg01.conf"

Old Volume Group configuration for "vg01" has been saved in
"/etc/lvmconf/vg01.conf.old"

Starting the modification by writing to all Physical Volumes

Applying the configuration to all Physical Volumes from

"/etc/lvmconf/vg01.conf"

Volume Group configuration has been restored to /dev/rdisk/c11t0d0

Volume Group configuration has been restored to /dev/rdisk/c11t1d0

Volume Group configuration has been restored to /dev/rdisk/c11t2d0

Volume Group configuration has been restored to /dev/rdisk/c11t3d0

Completed the modification process.

7. Activate the volume group.

```
# vgchange -a y -s vg01
Activated volume group
Volume group "vg01" has been successfully changed.
```

8. Check that the values are as requested.

```
# vgdisplay vg01
--- Volume groups ---
VG Name                /dev/vg01
VG Write Access        read/write
VG Status              available
Max LV                 255
Cur LV                1
Open LV                1
Max PV                 24
Cur PV                4
Act PV                 4
Max PE per PV         20988
VGDA                   8
PE Size (Mbytes)      4
Total PE               69456
Alloc PE               16
Free PE                69440
Total PVG              1
Total Spare PVs       0
Total Spare PVs in use 0
```


Changing a bootable disk to non-bootable

Run `vgmodify -B n -t <vgname> <PV to convert>` with and without `-n`. Run `vgmodify -B n -o -r <vgname> <PV to convert>`. Select the entry from the output of the three commands which meets the requirements. If a setting from the `-tn` table is to be used and this requires physical extent renumbering lower then run `vgmodify -B n -tnv <vgname> <PV to convert>` and check that no disks need the first physical extent to be made free. If they do then free-up this extent on each disk using `pvmove`, `lvreduce` and/or `lvremove`. Run `vgmodify -B n -r -e max_pe -p max_pv <vgname> <PV to convert>`, where `max_pe` and `max_pv` are the chosen values. Finally, when a maintenance window is available, deactivate the volume group and run `vgmodify` with the same options as before but without `-r` or, if using the `-o` settings, `-B n -o <vg name> <PV to convert>`.

1. First run table mode and note whether there are any bootable disks reported.

```
# vgmodify -tv vg02
```

```
Volume Group configuration for /dev/vg02 has been saved in
/etc/lvmconf/vg02.conf
```

```
Deactivating Volume Group "vg02"
```

```
Volume group "vg02" has been successfully deactivated.
```

```
Current Volume Group settings:
```

Max LV	255
Max PV	10
Max PE per PV	8956
PE Size (Mbytes)	4
VGRA Size (Kbytes)	752

```
VGRA space (Kbytes) on Physical Volumes with extents in use:
```

PV	current	-n
/dev/rdisk/c0t0d0	896	4096
/dev/rdisk/c1t0d0	768	768
Summary	768	768

The space available for the configuration data will be limited by having bootable disks. If this is not a boot VG then consider changing the type, using `-B n`, of the following bootable disks:

```
/dev/rdisk/c1t0d0
```

```
Volume Group optimized settings (no PEs renumbered):
```

max_pv(-p)	max_pe(-e)	Disk size (Mb)
2	45820	183280
3	30460	121840
* Entries deleted for brevity		
44	2044	8176
45	1808	7232
* Entries deleted for brevity		
182	264	1056
255	252	1008

Note that the space for the VGRA on `c1t0d0` is 768Kb regardless of whether `-n` is used or not. This is to be expected for a bootable disk. Note how this affects the summary and is limiting both `-n` and non `-n` sizes.

2. Table mode with the bootable disk now made non-boot.

```
# vgmodify -tv -B n vg02 /dev/rdisk/c1t0d0
```

```
Volume Group configuration for /dev/vg02 has been saved in
/etc/lvmconf/vg02.conf
```

```
Current Volume Group settings:
```

Max LV	255
Max PV	10
Max PE per PV	8956
PE Size (Mbytes)	4
VGRA Size (Kbytes)	752

```
VGRA space (Kbytes) on Physical Volumes with extents in use:
```

PV	current	-n
/dev/rdisk/c1t0d0	896	4096
Summary	896	4096

```
Volume Group optimized settings (no PEs renumbered):
```

max_pv(-p)	max_pe(-e)	Disk size (Mb)
2	53756	215025
3	35836	143345
* Entries deleted for brevity		
42	2556	10225
46	2300	9201
* Entries deleted for brevity		
213	296	1185
255	252	1009

Compare this table with the previous one to see how the bootable disk is limiting the values. This will be even more noticeable if `-tvn` options are used with `-B n`. From the table select the values desired. We'll select `-p 42` and `-e 2556`.

3. Review the mode for the new settings.

```
# vgmodify -r -v -p 42 -e 2556 -B n vg02 /dev/rdisk/c1t0d0
```

```
Volume Group configuration for /dev/vg02 has been saved in
/etc/lvmconf/vg02.conf
```

```
Current Volume Group settings:
```

Max LV	255
Max PV	10
Max PE per PV	8956
PE Size (Mbytes)	4
VGRA Size (Kbytes)	752

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

```
Physical Volume "/dev/rdisk/c1t0d0" the current and new settings
differ
```

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

```
New Volume Group settings:
```

Max LV	255
Max PV	42
Max PE per PV	2556
PE Size (Mbytes)	4
VGRA Size (Kbytes)	896

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

4. Deactivate in preparation for modification.

```
# vgchange -a n vg02
```

```
Volume group "vg02" has been successfully changed.
```

5. Change mode.

```
# vgmodify -v -p 42 -e 2556 -B n vg02 /dev/rdisk/c1t0d0
Performing "vgchange -a y -l -p -s vg02" to collect data
Activated volume group
Volume group "vg02" has been successfully activated.
Volume Group configuration for /dev/vg02 has been saved in
/etc/lvmconf/vg02.conf
Deactivating Volume Group "vg02"
Volume group "vg02" has been successfully deactivated.
```

Current Volume Group settings:

Max LV	255
Max PV	10
Max PE per PV	8956
PE Size (Mbytes)	4
VGRA Size (Kbytes)	752

The current and new Volume Group parameters differ.
Physical Volume "/dev/rdisk/c1t0d0" the current and new settings
differ
An update to the Volume Group IS required

New Volume Group settings:

Max LV	255
Max PV	42
Max PE per PV	2556
PE Size (Mbytes)	4
VGRA Size (Kbytes)	896

```
New Volume Group configuration for "vg02" has been saved in
/etc/lvmconf/vg02.conf"
Old Volume Group configuration for "vg02" has been saved in
/etc/lvmconf/vg02.conf.old"
Starting the modification by writing to all Physical Volumes
Applying the configuration to all Physical Volumes from
/etc/lvmconf/vg02.conf"
Volume Group configuration has been restored to /dev/rdisk/c0t0d0
Volume Group configuration has been restored to /dev/rdisk/c1t0d0
Completed the modification process.
New Volume Group configuration for "vg02" has been saved in
/etc/lvmconf/vg02.conf.old"
Volume group "vg02" has been successfully changed.
```

6. Activate volume group.

```
# vgchange -a y -s vg02
Activated volume group
Volume group "vg02" has been successfully changed.
```

7. Verify settings.

vgdisplay vg02

```
--- Volume groups ---
VG Name                /dev/vg02
VG Write Access        read/write
VG Status              available
Max LV                 255
Cur LV                1
Open LV               1
Max PV                 42
Cur PV                2
Act PV                2
Max PE per PV         2556
VGDA                   4
PE Size (Mbytes)      4
Total PE              2046
Alloc PE               1
Free PE               2045
Total PVG              0
Total Spare PVs       0
Total Spare PVs in use 0
```

vgcfgrestore -l -n vg02

```
Volume Group Configuration information in "/etc/lvmconf/vg02.conf"
VG Name /dev/vg02
---- Physical volumes : 2 ----
  /dev/rdisk/c0t0d0 (Non-bootable)
  /dev/rdisk/c1t0d0 (Non-bootable)
```

Optimizing the metadata space

Run `vgmodify -r -o <vgname>`. If the command reports an update to the volume group is required then during a maintenance window, deactivate it and run `vgmodify -o <vgname>`.

1. Run `vgmodify`.

```
# vgmodify -r -o vg05
```

```
Current Volume Group settings:
```

Max LV	255
Max PV	16
Max PE per PV	1023
PE Size (Mbytes)	4
VGRA Size (Kbytes)	208

```
New configuration requires "max_pes" are increased from 1023 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
```

Using `-o` the `max_pes` can be increased from 1023 to 6652 as the VGRA size can be increased from 208 to 896Kb.

2. Deactivate the volume group in preparation for the change.

```
# vgchange -a n vg05
```

```
Volume group "vg05" has been successfully changed.
```

3. Modify the volume group.

```
# vgmodify -o vg05
```

```
Current Volume Group settings:
```

Max LV	255
Max PV	16
Max PE per PV	1023
PE Size (Mbytes)	4
VGRA Size (Kbytes)	208

```
New configuration requires "max_pes" are increased from 1023 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

```
New Volume Group configuration for "vg05" has been saved in "/etc/lvmconf/vg05.conf"
```

```
Old Volume Group configuration for "vg05" has been saved in "/etc/lvmconf/vg05.conf.old"
```

```
Starting the modification by writing to all Physical Volumes
```

Applying the configuration to all Physical Volumes from
"/etc/lvmconf/vg05.conf"
Completed the modification process.
New Volume Group configuration for "vg05" has been saved in
"/etc/lvmconf/vg05.conf.old"
Volume group "vg05" has been successfully changed.

4. Activate the volume group.

```
# vgchange -a y -s vg05
Activated volume group
Volume group "vg05" has been successfully changed.
```

5. Check the volume group.

```
# vgdisplay vg05
--- Volume groups ---
VG Name                /dev/vg05
VG Write Access        read/write
VG Status              available
Max LV                 255
Cur LV                0
Open LV                1
Max PV                 16
Cur PV                2
Act PV                 2
Max PE per PV         6652
VGDA                   4
PE Size (Mbytes)      4
Total PE               2046
Alloc PE               0
Free PE                2046
Total PVG              0
Total Spare PVs       0
Total Spare PVs in use 0
```

Handling dynamic LUN expansion and configurations where not all the disk space can be allocated

Run `vgmodify -v -r <vgname>`. Any disks which have been resized since `pvcreate` was run on them are reported like this:

```
/dev/rdisk/c0t0d0" size changed from 16777216 to 100000000kb
```

Also note if there messages telling you there is insufficient space to allocate all the disk space on any device e.g.:

```
/dev/rdisk/c0t4d2 Warning: Max_PE_per_PV for the volume group (4348) too small for this PV (24413).
```

```
Using only 4348 PEs from this physical volume.
```

If you wish to change the volume group parameters then follow the guidance given in the previous sections. Otherwise, during a maintenance window, deactivate the volume group and run `vgmodify -v <vgname>`.

1. Review mode.

```
# vgmodify -r -v vg04
```

```
Volume Group configuration for /dev/vg04 has been saved in /etc/lvmconf/vg04.conf
```

```
Current Volume Group settings:
```

Max LV	255
Max PV	16
Max PE per PV	1023
PE Size (Mbytes)	4
VGRA Size (Kbytes)	208

```
/dev/rdisk/c5t0d0 Warning: Max_PE_per_PV for the volume group (1023) too small for this PV (1464).
```

```
Using only 1023 PEs from this physical volume.
```

```
/dev/rdisk/c6t0d0 Warning: Max_PE_per_PV for the volume group (1023) too small for this PV (1220).
```

```
Using only 1023 PEs from this physical volume.
```

```
"/dev/rdisk/c5t0d0" size changed from 4194304 to 6000000kb
```

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

```
New Volume Group settings:
```

Max LV	255
Max PV	16
Max PE per PV	1023
PE Size (Mbytes)	4
VGRA Size (Kbytes)	208

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

`c5t0d0` has been increased in size since it was initialized via `pvcreate`. `c6t0d0` has its correct size recorded by LVM so has not changed size since `pvcreate`. Both disks are unable to allocate all their space as the maximum number of PEs per PV setting is too low. A setting of 1464 will accommodate the largest disk currently in the volume group (`c5t0d0`). Let's take a look at the table mode and see what configurations would suit.

```
# vgmodify -t -v vg04
```

```
Volume Group configuration for /dev/vg04 has been saved in  
/etc/lvmconf/vg04.conf
```

```
Current Volume Group settings:
```

```
Max LV      255  
Max PV      16  
Max PE per PV 1023  
PE Size (Mbytes) 4  
VGRA Size (Kbytes) 208
```

```
VGRA space (Kbytes) on all Physical Volumes:
```

```
PV          current      -n  
/dev/rdisk/c4t0d0      896      4096  
/dev/rdisk/c5t0d0      896      4096  
/dev/rdisk/c6t0d0      896      4096  
Summary                896      4096
```

```
Volume Group optimized settings (no PEs renumbered):
```

```
max_pv(-p)    max_pe(-e)    Disk size (Mb)  
3             35836        143345  
4             26876        107505  
* Entries deleted for brevity  
20           5372         21489  
21           5116         20465  
* Entries deleted for brevity  
60           1788         7153  
70           1532         6129  
84           1276         5105  
* Entries deleted for brevity
```

2. Max_pv values up to 70 provide for a sufficient number of max_pes for the current largest disk. We'll select the entry for max_pv of 20.

```
# vgmodify -r -p 20 -e 5372 -v vg04
```

```
Volume Group configuration for /dev/vg04 has been saved in  
/etc/lvmconf/vg04.conf
```

```
Current Volume Group settings:
```

```
Max LV      255  
Max PV      16  
Max PE per PV 1023  
PE Size (Mbytes) 4  
VGRA Size (Kbytes) 208
```

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

```
"/dev/rdisk/c5t0d0" size changed from 4194304 to 6000000kb
```

```
Physical Volume "/dev/rdisk/c6t0d0" the current and new settings  
differ
```

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

```
New Volume Group settings:
```

```
Max LV      255  
Max PV      20  
Max PE per PV 5372  
PE Size (Mbytes) 4  
VGRA Size (Kbytes) 896
```

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


3. Deactivate in preparation for the modification.

```
# vgchange -a n vg04
Volume group "vg04" has been successfully changed.
```

4. Change mode.

```
# vgmodify -p 20 -e 5372 -v vg04
Performing "vgchange -a y -l -p -s vg04" to collect data
Activated volume group
Volume group "vg04" has been successfully activated.
Volume Group configuration for /dev/vg04 has been saved in
/etc/lvmconf/vg04.conf
Deactivating Volume Group "vg04"
Volume group "vg04" has been successfully deactivated.
```

Current Volume Group settings:

Max LV	255
Max PV	16
Max PE per PV	1023
PE Size (Mbytes)	4
VGRA Size (Kbytes)	208

The current and new Volume Group parameters differ.

"/dev/rdisk/c5t0d0" size changed from 4194304 to 6000000kb

Physical Volume "/dev/rdisk/c6t0d0" the current and new settings differ

An update to the Volume Group IS required

New Volume Group settings:

Max LV	255
Max PV	20
Max PE per PV	5372
PE Size (Mbytes)	4
VGRA Size (Kbytes)	896

New Volume Group configuration for "vg04" has been saved in
"/etc/lvmconf/vg04.conf"

Old Volume Group configuration for "vg04" has been saved in
"/etc/lvmconf/vg04.conf.old"

Starting the modification by writing to all Physical Volumes
Applying the configuration to all Physical Volumes from
"/etc/lvmconf/vg04.conf"

Volume Group configuration has been restored to /dev/rdisk/c4t0d0

Volume Group configuration has been restored to /dev/rdisk/c5t0d0

Volume Group configuration has been restored to /dev/rdisk/c6t0d0

Completed the modification process.

New Volume Group configuration for "vg04" has been saved in
"/etc/lvmconf/vg04.conf.old"

Volume group "vg04" has been successfully changed.

5. Activate the volume group.

```
# vgchange -a y -s vg04
Activated volume group
Volume group "vg04" has been successfully changed.
```

6. Check the settings.

```
# vgdisplay vg04
--- Volume groups ---
VG Name                /dev/vg04
VG Write Access        read/write
VG Status              available
Max LV                 255
Cur LV                0
Open LV               11
Max PV                 20
Cur PV                3
Act PV                3
Max PE per PV         5372
VGDA                  6
PE Size (Mbytes)      4
Total PE              3707
Alloc PE              0
Free PE               3707
Total PVG             0
Total Spare PVs       0
Total Spare PVs in use 0

# pvdisplay /dev/dsk/c5t0d0
--- Physical volumes ---
PV Name                /dev/dsk/c5t0d0
VG Name                /dev/vg04
PV Status              available
Allocatable           yes
VGDA                  2
Cur LV                0
PE Size (Mbytes)      4
Total PE              1464
Free PE               1464
Allocated PE          0
Stale PE              0
IO Timeout (Seconds)  default
Autoswitch            On
```

Note the Total PE is now at 1464 which is greater than the previous max_pe setting of 1023.

```
# pvdisplay /dev/dsk/c6t0d0
--- Physical volumes ---
PV Name                /dev/dsk/c6t0d0
VG Name                /dev/vg04
PV Status              available
Allocatable           yes
VGDA                  2
Cur LV                0
PE Size (Mbytes)      4
Total PE              1220
Free PE               1220
Allocated PE          0
Stale PE              0
IO Timeout (Seconds)  default
Autoswitch            On
```

Note the Total PE is now at 1220.

LVM `vgmodify` Messages

`vgmodify(1M)`

1. Developing a new set of PVIDs. This will take approximately 10 seconds

When `vgmodify` makes any changes to the volume group, a new physical volume identifier (PVID) is created for each physical volume. This can take one second per physical volume in the volume group. A new PVID helps to ensure that a partial restoration from a configuration backup file created before this modification will not lead to a mixed configuration.

2. `"/dev/rdisk/c2t3d0"` size changed from 768000 to 8700000kb

`vgmodify` has detected that the size of `c2t3d0` has changed. When the physical volume was created (see `pvcreate(1M)`) it was 768,000kb it is now 8,700,000kb. This detail is reported when using the `-v` argument.

3. Expected 5 Physical Volumes but only found 4

The volume group has five physical volumes but `vgmodify` could only locate four of these. Check to make sure all five physical volumes are accessible. `vgmodify` cannot continue until it can access all five.

4. VGRA space (Kbytes) without PE renumbering

This is the amount of space available for the VGRA when physical extent renumbering is left unchanged.

5. VGRA space (Kbytes) PE renumbering lower

This is the amount of space available for the VGRA when physical extents are renumbered lower so that the first physical extent can be used for metadata. All physical extent numbers will be reduced by one.

6. VGRA space (Kbytes) PE renumbering higher

This is the amount of space available for the VGRA when physical extents are renumbered higher so that metadata space can be returned to user space. All physical extent numbers will be increased by one.

7. Prior to modification test IO to `"/dev/rdisk/c2t3d0"` failed

Before making any modifications a test write is performed to each physical volume. The test write failed to this device. `vgmodify` has stopped before making any changes applicable to the new configuration. Check the device before repeating the command.

8. Physical Volume `"/dev/rdisk/c2t3d0"` is already the type requested

The `-B` option has been used with `c2t3d0` in the list of physical volumes but this physical volume is already the type requested. i.e. It is already bootable if `-B y`, or non-bootable if `-B n`.

9. Installing the new configuration was partially successful. Examine `vgcfgrestore(1M)` messages to understand which disks may require manual restoration from the `"/etc/lvmconf/vg01.conf"` file. The Volume Group should activate in its current state without quorum override but not all disks may attach.

`vgmodify` was partially successfully in placing the new configuration onto the physical volumes but some failed. There should be errors messages from `vgcfgrestore` prior to this message. Check the disks associated with the failures and then manually restore to these disks.

10. The space available for the configuration data will be limited by having bootable disks. If this is not a boot VG then consider changing the type, using `-B n`, of the following bootable disks:
`/dev/rdisk/c2t3d0`
`/dev/rdisk/c3t4d0`

Bootable disks (that is disks that were created using `pvcreate -B` or changed type using `vgmodify -B y`) limit the space available for the VGRA. If this is not a boot volume group then use `-B n` to change all the disks in the list to non-boot.

11. Physical Extent zero is not free on all PVs. You will not achieve these values until the first extent is made free (see `pvmove(1M)`) on all the following disks:
`/dev/rdisk/c2t4d0`

`vgmodify` has been run with `-n` and it wants to lower the physical extent numbers on `c2t4d0` but to do this the first extent must be free so that its space can be given over to metadata.

12. VGRA for the disk is too big for the specified parameters. Decrease `max_PVs` and/or `max_PEs`.

The selected `max_pv` and `max_pe` values will create a VGRA that is too large for the space available on all physical volumes. The VGRA cannot be greater than the physical extent size.

For More Information

To learn more about LVM and HP-UX system administration, refer to the following documents on the HP documentation website (<http://docs.hp.com>):

- HP-UX System Administrator's Guide: Logical Volume Configuration
<http://docs.hp.com/en/5991-6481/5991-6481.pdf>
- LVM White Papers:
 - LVM Limits
http://docs.hp.com/en/6054/Limits_wp.pdf
 - LVM Migration from Legacy to Agile Naming Model
http://docs.hp.com/en/LVMmigration1/LVM_Migration_to_Agile.pdf
 - LVM New Features in HP-UX 11i v3
http://docs.hp.com/en/LVM-11iv3features/LVM_New_Features_11iv3.pdf
 - LVM Online Disk Replacement (LVM OLR)
http://docs.hp.com/en/7161/LVM_OLR_whitepaper.pdf
 - LVM Volume Group Quiesce/Resume
http://docs.hp.com/en/lvm-quiesce/LVM_Quiesce-Resume.pdf
 - SLVM Single-Node Online Reconfiguration (SLVM SNOR)
http://docs.hp.com/en/7389/LVM_SNOR_whitepaper.pdf
 - When Good Disks Go Bad: Dealing with Disk Failures under LVM
http://docs.hp.com/en/5991-1236/When_Good_Disks_Go_Bad.pdf

Call to action

HP welcomes your input. Please give us comments about this white paper, or suggestions for LVM or related documentation, through our technical documentation feedback website:

<http://docs.hp.com/en/feedback.html>