

# HP-UX LVM Supported Limits



## Table of Contents:

Abstract .....	2
Introduction .....	2
Supported Limits for Version 1.0 Volume Groups .....	3
Calculating an Optimal Extent Size for a Version 1.0 Volume Group.....	4
Sample Shell Script.....	4
Supported Limits for Version 2.0 Volume Groups .....	6
Calculating an Optimal Size for a Version 2.0 Volume Group.....	7
Determining LVM's Maximum Limits on a System.....	8
Glossary .....	9
For More Information.....	10
Call to Action .....	10

# Abstract

Starting with HP-UX 11i v2 (11.23), the supported limits of LVM volume groups have been gradually increasing. This following whitepaper summarizes the supported limits for the March 2008 release of HP-UX 11i v3 (11.31).

It is assumed that the reader has a basic knowledge of LVM.

The document is intended for system administrators, operators, and customers who are creating or managing volume groups.

## Introduction

With the introduction of the March 2008 release of HP-UX 11i v3 (11.31), LVM supports a new volume group type referred to as a Version 2.0 volume group. This paper is structured into two parts with the first part covering the original Version 1.0 volume group and the second part covering Version 2.0 volume groups. For details about all the differences and how to use Version 2.0 volume groups, see the "LVM Version 2.0 Volume Groups in HP-UX 11i v3" white paper, available at <http://docs.hp.com/en/oshpux11iv3.html#LVM%20Volume%20Manager>.

## Supported Limits for Version 1.0 Volume Groups

The following table summarizes the supported limits in LVM for HP-UX releases from 11.00 onwards. Unless noted otherwise, the values are the same across releases.

Parameter	Operation to change the value	Min Value	Default Value	Max Value
Max number of Version 1.0 VGs per system  Note: in 11i v3, tunable was removed	Kernel tunable: <i>kctune</i> maxvgs='value'	0	10	256
Max LVs Per VG <sup>1</sup>	<i>vgcreate</i> -l 'max_lv'	1	255	255
Max PVs per VG <sup>1</sup>	<i>vgcreate</i> -p 'max_pv'	1	16	255
Max Extents per PV <sup>1</sup>  The default value may get adjusted to (Disk space/ Extent size) of the first PV used for creating the VG, if that value is greater than 1016.	<i>vgcreate</i> -e 'max_pe'	1	1016	65535
Extent Size	<i>vgcreate</i> -s 'pe_size'	1 MB	4 MB	256 MB
Effective size of a PV <sup>1</sup>  Following or superseding patches are required for supporting disks larger than 256GB: 11.00 – PHKL_30553 11.11 – PHKL_30622 11.23 – PHKL_31500	<i>pvcreate</i> -s 'disk_size'	Extent Size	LUN Capacity	2 TB
Size of a LV 11.11 & 11.23	<i>lvcreate/lvextend</i> -l 'le_number'   -L 'lv_size'	0	0	2 TB  [I/Os beyond 2TB will be returned with error]
Size of a LV 11.31				16 TB
Mirror copies per LV	<i>lvcreate/lvextend</i> -m 'mirror_copies'	0	0	2
Stripes of LV	<i>lvcreate</i> -i 'stripes'	2	None – Must specify	Max PVs per VG
Stripe size of LV	<i>lvcreate</i> -l 'stripe_size'	4KB	8KB	32768KB

<sup>1</sup> With 11.31, the *vgmodify* command enables changing these values on an existing VG. For more information, see the *vgmodify(1M)* man page and release notes.

## Calculating an Optimal Extent Size for a Version 1.0 Volume Group

Sometimes when creating a Version 1.0 volume group (VG), the *vgcreate* command may abort with a message that the extent size is too small (too big error or with newer patches a more informative error explaining that the VGRA is too big). In this situation the user is expected to manually increase the extent size and re-issue the *vgcreate* command.

Increasing the extent size increases the data area marked stale when a write to a mirrored logical volume (LV) fails and that can increase the time required for resynchronizing the stale data. Also, more space than intended may be allocated to the LV because the space is allocated in units of extent size. Therefore, the optimal extent size is the smallest value that can be used to successfully create the volume group with the given configuration parameters.

The minimum extent size for a VG is calculated using the maximum number of logical volumes (MAXLVs) and physical volumes (MAXPVs) in the VG and the maximum number of physical extents (MAXPXs) per each physical volume (PV).

For a VG with bootable PVs, the metadata must fit within 768 KB. Therefore, a *vgcreate* command with a set of values for MAXLVs, MAXPVs and MAXPXs that succeed on a VG without bootable PVs may fail on a VG with bootable PVs. In this situation, if you must add a bootable PV to a VG, recreate the VG by giving lesser values for these arguments. By far the biggest factor in the size of the metadata is the values for MAXPVs and MAXPXs. Alternatively, convert the bootable PV to a normal PV by rerunning *pvcreate* on that PV without the '-B' option and then add it to the VG. For a PV that is included in a VG, you can use *vgmodify* to change a PV from bootable to a normal PV.

## Sample Shell Script

The following shell script creates and compiles a small program that gives the minimum extent size for a given VG:

```
#!/usr/bin/sh
cat << EOF > vgrasize.c
#include <stdio.h>

#define BS 1024 /* Device block Size */
#define roundup(val, rnd) (((val + rnd - 1) / rnd) * rnd)

main(int argc, char *argv[])
{
    int i, length, lvs, pvs, pxs;

    if (argc != 4) {

        /* Usage example:
        *   Maximum LVs in the VG = 255
        *   Maximum PVs in the VG = 16
        *   Maximum extents per PV = 2500
        *
        *   $ vgrasize 255 16 2500
        */
        printf("USAGE: %s <MAXLVs> <MAXPVs> <MAXPXs>\n", argv[0]);
        exit(1);
    }
    lvs = atoi(argv[1]); pvs = atoi(argv[2]); pxs = atoi(argv[3]);
```

```

length = 16 + 2 * roundup(2 +
    (roundup(36 + ((3 * roundup(pvs, 32)) / 8) +
        (roundup(pxs, 8) / 8) * pvs, BS) +
    roundup(16 * lvs, BS) +
    roundup(16 + 4 * pxs, BS) * pvs) / BS, 8);

if (length > 768) {
printf("Warning: A bootable PV cannot be added to a VG \n"
    "created with the specified argument values. \n"
    "The metadata size %d Kbytes, must be less \n"
    "than 768 Kbytes.\n"
    "If the intention is not to have a boot disk in this \n"
    "VG then do not use '-B' option during pvcreate(1M) \n"
    "for the PVs to be part of this VG. \n", length);
}

length = roundup(length, 1024) / 1024;

if (length > 256) {
printf("Cannot configure a VG with the maximum values"
    " for LVs, PVs and PXs\n");
exit(1);
}

for (i = 1; i < length ; i = i << 1) { }

printf("\nMinimum extent size for this configuration = %d MB\n", i);
}
EOF
make vgrasize

```

## Supported Limits for Version 2.0 Volume Groups

The following table summarizes the supported limits in LVM for Version 2.0 VGs. Unless noted otherwise, the values are the same across releases. Note, creation of Version 2.0 VGs and the inherent limitations are very different from Version 1.0 VGs. For more information, see the “LVM 2.0 Volume Groups in HP-UX 11i v3” white paper available at <http://docs.hp.com/en/oshpux11iv3.html#LVM%20Volume%20Manager>.

Parameter	Operation to change the value	Min Value	Default Value	Max Value
Max number of Version 2.0 VGs per system	None – not needed	0	n/a	512
Max LVs Per VG	None – not needed	511	511	511
Max PVs per VG	None – not needed	511	511	511
Max VG size	<i>vgcreate -S 'max_vgsize'</i>	1MB <sup>2</sup>	None – Must specify	2PB
Max size of a PV		Extent Size	LUN Capacity	16 TB
Max Extents per PV	None – not needed	1	LUN capacity/ extent size	16384
Extent Size	<i>vgcreate -s 'pe_size'</i>	1 MB	None – Must specify	256 MB
Size of a LV	<i>lvcreate/lvextend -L 'lv_size'</i>	0	0	256 TB
Max Extents per LV	<i>lvcreate/lvextend -l 'le_number'</i>	0	0	33554432
Mirror copies per LV	<i>lvcreate/lvextend -m 'mirror_copies'</i>	0	0	5
Stripes of LV	<i>lvcreate -i 'stripes'</i>	2	None – Must specify	Max PVs per VG
Stripe size of LV	<i>lvcreate -l 'stripe_size'</i>	4KB	None – Must specify	262144 KB

<sup>2</sup> If the disk is larger than the max VG size specified, *vgcreate* adjusts the minimum VG size to match the disk size.

## Calculating an Optimal Size for a Version 2.0 Volume Group

Like Version 1.0 VGs, there is a relationship between extent size and maximum VG size. There is also a limitation on the number of extents an individual VG can contain. To aid in determining the proper size, the `vgcreate` command has a new option `-E` which returns the maximum sizes based on either the physical extent size or Max VG size.

### Example

Once you know the VG size you want to provision for, use `vgcreate` with the `-E` option to determine the minimum extent size required to achieve it.

What is the minimum extent size to provision a VG for 1 PB?

```
# vgcreate -V 2.0 -E -S 1p  
Max_VG_size=1p:extent_size=32m
```

What is the maximum VG size with an extent size of 16MB?

```
# vgcreate -V 2.0 -E -s 16  
Max_VG_size=512t:extent_size=16m
```

## Determining LVM's Maximum Limits on a System

With the March 2008 release of HP-UX 11i v3 (11.31), a new command has been introduced that enables the system administrator to determine the maximum limits LVM supports on the target system. The *lvmadm* command displays the implemented limits for Version 1.0 and Version 2.0 VGs. It is not possible to create a VG that exceeds these limits.

### Example

```
# 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           3354432
Max PXs per PV           16777216
Max Extent Size (Mbytes) 256
```

# Glossary

**KB**

A kilobyte unit of information equal to  $2^{10}$  or 1024 bytes

**MB**

A megabyte unit of information equal to  $2^{20}$  or 1,048,576 bytes

**GB**

A gigabyte unit of information equal to  $2^{30}$  or 1,073,741,824 bytes

**TB**

A terabyte unit of information equal to  $2^{40}$  or 1,099,511,627,776 bytes

**PB**

A petabyte unit of information equal to  $2^{50}$  or 1,125,899,906,842,624 bytes

## For More Information

To learn more about LVM features, see the following documents on the 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 Version 2.0 Volume Groups in HP-UX 11i v3
- LVM Online Disk Replacement (LVM OLR)
- LVM Volume Group Dynamic LUN expansion (DLE)/vgmodify
- LVM Volume Group Quiesce/Resume
- SLVM Single-Node Online Reconfiguration (SLVM SNOR)
- When Good Disks Go Bad: Dealing with Disk Failures under LVM

To learn more about configuring LVM and migration of LVM VG configuration from legacy to agile naming model, see the following documents on the 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
- HP-UX LVM Performance Assessment (The whitepaper will be available soon)

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

- Overview: The Next Generation Mass Storage Stack

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

- Node and Host Name Sizes on HP-UX: Using the Expanded Capabilities of 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.