

# [RHEL4.7 RHEL5.3] "vgreduce --removemissing --force" does not work when a PV is missing that was part of an LV

---

## Issue

- Unable to remove a volume group with missing PVs
- "vgreduce --removemissing --force" does not restore volume group to consistent state

## Environment

- Red Hat Enterprise Linux 4 or 5 with an lvm2 release prior to 2.02.46
  - Red Hat Enterprise Linux 4.7 (lvm2-2.02.42-5.el4\_8.3-x86\_64)
  - Red Hat Enterprise Linux 5.3 or below
- A volume group containing LVs with PVs missing

## Resolution

WARNING: The following procedure(s) will restore the volume group to a consistent state, but must be done carefully.

There are two known ways to restore the volume group to a consistent state:

1. If the volume group contains other LVs / PVs that need to remain in tact, then use option 2.
2. Otherwise, the volume group can be destroyed by re-initializing each of the remaining PVs as follows:

- Identify the devices that correspond to the missing PVs.
- Wipe the devices that correspond to the missing PVs with either "pvremove -ff" or "pvcreate -ff"

2. Edit the last known metadata archive and use vgcfgrestore

- Locate the last metadata 'archive' file for the volume group in question in /etc/lvm/archive. The latest archive file for each volume group will be the one with the highest number on the filename.
- Copy the last metadata archive file to a temporary file name, "myvg-temp".
- Use a text editor to search for all occurrences of "MISSING" in the file. Make a note of the "pvNN" number.
- Search for "pvNN" found in previous step. Remove all physical\_volume sections beginning with "pvNN {", as well as all logical volume sections which include "pvNN" as a segment.
- Save the edited file.
- Run vgcfgrestore to restore "myvg": vgcfgrestore -f myvg-temp myvg

[RHEL4.7 RHEL5.3] "vgreduce --removemissing --force" does not work when a PV is missing that was part of an LV

## Root Cause

This is a known bug.

## Diagnostic Steps

1. Examine the latest vg metadata backup file in /etc/lvm/archive. Open the file with an editor, and search for PVs with the "MISSING" flag.

Look for any duplicate PVs (ones with the same "pvNN" name, but different UUIDs) in the file. For example:

```
pv96 {  
  
    id = "sokfZx-wIOU-pZ5e-isiP-rnjE-pG7I-W8Lfar"  
  
    device = "unknown device"    # Hint only  
  
  
    status = ["ALLOCATABLE"]  
  
    flags = ["MISSING"]  
  
    dev_size = 70709216    # 33.7168 Gigabytes  
  
    pe_start = 384  
  
    pe_count = 8631 # 33.7148 Gigabytes  
  
}
```

```
pv96 {
```

[RHEL4.7 RHEL5.3] "vgreduce --removemissing --force" does not work when a PV is missing that was part of an LV

```
id = "nAE7MC-Lids-Nu3L-kYiU-2y2U-Xkqy-5BuuJC"

device = "unknown device"    # Hint only

status = ["ALLOCATABLE"]

flags = ["MISSING"]

dev_size = 70709216    # 33.7168 Gigabytes

pe_start = 384

pe_count = 8631 # 33.7148 Gigabytes

}
```

2. Attempt the following normal sequence to cleanup, but this does not result in a consistent VG.

```
vgreduce --removemissing --force local_3par-dg
```

```
vgreduce --removemissing local_3par-dg
```

```
vgremove local_3par-dg
```

3. Even after using "vgreduce --removemissing --force" lvm reporting commands such as pvs, lvs, vgs show messages like:

```
LV scratch3: segment 1 has inconsistent PV area 0
```

```
Couldn't read all logical volumes for volume group local_3par-dg.
```

```
Couldn't find device with uuid '3dxqCf-ZtmU-oIUj-UO8e-7JAY-AEHX-GO8ZIN'.
```

[RHEL4.7 RHEL5.3] "vgreduce --removemissing --force" does not work when a PV is missing that was part of an LV

4. An upgrade to the latest RHEL4 lvm2 and device-mapper RPMs, lvm2-2.02.42-9.el4 and device-mapper-1.02.28-2.el4\_8.1 does not solve the problem.