NAME

pvck − Check metadata on physical volumes

SYNOPSIS

pvck *option_args position_args*
[*option_args*]

−−commandprofile *String*
−−config *String*
−d|**−−debug**
−−driverloaded y|**n**
−−dump headers|**metadata**|**metadata_all**|**metadata_search**
−f|**−−file** *String*
−h|**−−help**
−−labelsector *Number*
−−lockopt *String*
−−longhelp
−−nolocking
−−profile *String*
−−[pv]metadatacopies 0|**1**|**2**
−q|**−−quiet**
−−repair
−−repairtype pv_header|**metadata**|**label_header**
−−settings *String*
−t|**−−test**
−v|**−−verbose**
−−version
−y|**−−yes**

DESCRIPTION

pvck checks and repairs LVM metadata on PVs.

Dump options

headers

Print LVM on-disk headers and structures: label_header, pv_header, mda_header(s), and metadata text. Warnings are printed if any values are incorrect. The label_header and pv_header both exist in a 512 byte sector, usually the second sector of the device. An mda_header exists in a 512 byte sector at offset 4096 bytes. A second mda_header can optionally exist near the end of the device. The metadata text exists in an area (about 1MiB by default) immediately following the mda_header sector. The metadata text is checked but not printed (see other options).

metadata

Print the current LVM VG metadata text (or save to a file), using headers to locate the latest copy of metadata. If headers are damaged, metadata may not be found (see metadata_search). Use −−settings "mda_num=2" to look in mda2 (the second mda at the end of the device, if used). The metadata text is printed to stdout or saved to a file with −−file.

metadata_all

List all versions of VG metadata found in the metadata area, using headers to locate metadata. Full copies of all metadata are saved to a file with the −−file option. If headers are damaged, metadata may not be found (see metadata_search). Use −−settings "mda_num=2" as above. Use −v to include descriptions and dates when listing metadata versions.

metadata_search

List all versions of VG metadata found in the metadata area, searching common locations so metadata can be found if headers are damaged. Full copies of all metadata are saved to a file with the −−file option. To

save one specific version of metadata, use −−settings "metadata_offset=<offset>", where the offset is taken from the list of versions found. Use −v to include descriptions and dates when listing metadata versions.

metadata_area
Save the entire text metadata area to a file without processing.

Repair options

−−repair
Repair headers and metadata on a PV. This uses a metadata input file that was extracted by −−dump, or a backup file (from /etc/lvm/backup). When possible, use metadata saved by −−dump from another PV in the same VG (or from a second metadata area on the PV).

There are cases where the PV UUID needs to be specified for the PV being repaired. It is specified using −−settings "pv_uuid=<UUID>". In particular, if the device name for the PV being repaired does not match the previous device name of the PV, then LVM may not be able to determine the correct PV UUID. When headers are damaged on more than one PV in a VG, it is important for the user to determine the correct PV UUID and specify it in −−settings. Otherwise, the wrong PV UUID could be used if device names have been swapped since the metadata was last written.

If a PV has no metadata areas and the pv_header is damaged, then the repair will not know to create no metadata areas during repair. It will by default repair metadata in mda1. To repair with no metadata areas, use -−settings "mda_offset=0 mda_size=0".

There are cases where repair should be run on all PVs in the VG (using the same metadata file): if all PVs in the VG are damaged, if using an old metadata version, or if a backup file is used instead of raw metadata (taken from pvck dump.)

Using −−repair is equivalent to running −−repairtype pv_header followed by -−repairtype metadata.

−−repairtype pv_header
Repairs the header sector, containing the pv_header and label_header.

−−repairtype metadata
Repairs the mda_header and metadata text. It requires the headers to be correct (having been undamaged or already repaired).

−−repairtype label_header
Repairs label_header fields, leaving the pv_header (in the same sector) unchanged. (repairtype pv_header should usually be used instead.)

Settings

The −−settings option controls or overrides certain dump or repair behaviors. All offset and size values in settings are in bytes (units are not recognized.) These settings are subject to change.

mda_num=1|2
Select which metadata area should be used. By default the first metadata area (1) is used. mda1 is always located at offset 4096. mda2, at the end of the device, often does not exist (it's not created by default.) If mda1 is erased, mda2, if it exists, will often still have metadata.

metadata_offset=*bytes*
Select metadata text at this offset. Use with metadata_search to print/save one instance of metadata text.

mda_offset=*bytes* **mda_size=***bytes*
Refers to a metadata area (mda) location and size. An mda includes an mda_header and circular metadata

text buffer. Setting this forces metadata_search look for metadata in the given area instead of the standard locations. When set to zero with repair, it indicates no metadata areas should exist.

mda2_offset=*bytes* **mda2_size=***bytes*
When repairing a pv_header, this forces a specific offset and size for mda2 that should be recorded in the pv_header.

pv_uuid=*uuid*
Specify the PV UUID of the device being repaired. When not specified, repair will attempt to determine the correct PV UUID by matching a device name in the metadata.

device_size=*bytes*
data_offset=*bytes*
When repairing a pv_header, the device_size, data_offset, and pvid can all be specified directly, in which case these values are not taken from a metadata file (where they usually come from), and the metadata file can be omitted. data_offset is the starting location of the first physical extent (data), which follows the first metadata area.

USAGE

Check for metadata on a device

pvck *PV ...*
[COMMON_OPTIONS]
-

Check and print LVM headers and metadata on a device

pvck −−dump headers|**metadata**|**metadata_all**|**metadata_search** *PV*
[**−f**|**−−file** *String*]
[**−−settings** *String*]
[**−−[pv]metadatacopies 0**|**1**|**2**]
[COMMON_OPTIONS]
-

Repair LVM headers or metadata on a device

pvck −−repairtype pv_header|**metadata**|**label_header** *PV*
[**−f**|**−−file** *String*]
[**−−settings** *String*]
[COMMON_OPTIONS]
-

Repair LVM headers and metadata on a device

pvck −−repair −f|**−−file** *String PV*
[**−−settings** *String*]
[COMMON_OPTIONS]
-

Common options for command:
[**−−labelsector** *Number*]

Common options for lvm:
[**−d**|**−−debug**]
[**−h**|**−−help**]
[**−q**|**−−quiet**]

[**−t**|**−−test**]
[**−v**|**−−verbose**]
[**−y**|**−−yes**]
[**−−commandprofile** *String*]
[**−−config** *String*]
[**−−driverloaded y**|**n**]
[**−−lockopt** *String*]
[**−−longhelp**]
[**−−nolocking**]
[**−−profile** *String*]
[**−−version**]

OPTIONS

−−commandprofile *String*
The command profile to use for command configuration. See **lvm.conf**(5) for more information about profiles.

−−config *String*
Config settings for the command. These override lvm.conf settings. The String arg uses the same format as lvm.conf, or may use section/field syntax. See **lvm.conf**(5) for more information about config.

−d|**−−debug** ...
Set debug level. Repeat from 1 to 6 times to increase the detail of messages sent to the log file and/or syslog (if configured).

−−driverloaded y|**n**
If set to no, the command will not attempt to use device-mapper. For testing and debugging.

−−dump headers|**metadata**|**metadata_all**|**metadata_search**
Dump headers and metadata from a PV for debugging and repair. Option values include: **headers** to print and check LVM headers, **metadata** to print or save the current text metadata, **metadata_all** to list or save all versions of metadata, **metadata_search** to list or save all versions of metadata, searching standard locations in case of damaged headers, **metadata_area** to save an entire text metadata area to a file.

−f|**−−file** *String*
Metadata file to read or write.

−h|**−−help**
Display help text.

−−labelsector *Number*
By default the PV is labelled with an LVM2 identifier in its second sector (sector 1). This lets you use a different sector near the start of the disk (between 0 and 3 inclusive − see LABEL_SCAN_SECTORS in the source). Use with care.

−−lockopt *String*
Used to pass options for special cases to lvmlockd. See **lvmlockd**(8) for more information.

−−longhelp
Display long help text.

−−nolocking
Disable locking.

−−profile *String*
An alias for −−commandprofile or −−metadataprofile, depending on the command.

−−[pv]metadatacopies 0|**1**|**2**
The number of metadata areas to set aside on a PV for storing VG metadata. When 2, one copy of the VG metadata is stored at the front of the PV and a second copy is stored at the end. When 1,

one copy of the VG metadata is stored at the front of the PV. When 0, no copies of the VG metadata are stored on the given PV. This may be useful in VGs containing many PVs (this places limitations on the ability to use vgsplit later.)

−q|**−−quiet** ...
Suppress output and log messages. Overrides −−debug and −−verbose. Repeat once to also suppress any prompts with answer 'no'.

−−repair
Repair headers and metadata on a PV.

−−repairtype pv_header|**metadata**|**label_header**
Repair headers and metadata on a PV. See command description.

−−settings *String*
Specifies command specific settings in "Key = Value" form. Combine multiple settings in quotes, or repeat the settings option for each.

−t|**−−test**
Run in test mode. Commands will not update metadata. This is implemented by disabling all metadata writing but nevertheless returning success to the calling function. This may lead to unusual error messages in multi-stage operations if a tool relies on reading back metadata it believes has changed but hasn't.

−v|**−−verbose** ...
Set verbose level. Repeat from 1 to 4 times to increase the detail of messages sent to stdout and stderr.

−−version
Display version information.

−y|**−−yes**
Do not prompt for confirmation interactively but always assume the answer yes. Use with extreme caution. (For automatic no, see −qq.)

VARIABLES

PV
Physical Volume name, a device path under /dev. For commands managing physical extents, a PV positional arg generally accepts a suffix indicating a range (or multiple ranges) of physical extents (PEs). When the first PE is omitted, it defaults to the start of the device, and when the last PE is omitted it defaults to end. Start and end range (inclusive): *PV*[**:***PE*−*PE*]... Start and length range (counting from 0): *PV*[**:***PE*+*PE*]...

String
See the option description for information about the string content.

Size[UNIT]
Size is an input number that accepts an optional unit. Input units are always treated as base two values, regardless of capitalization, e.g. 'k' and 'K' both refer to 1024. The default input unit is specified by letter, followed by |UNIT. UNIT represents other possible input units:**bBsSkKmMgGtTpPeE**. b|B is bytes, s|S is sectors of 512 bytes, k|K is KiB, m|M is MiB, g|G is GiB, t|T is TiB, p|P is PiB, e|E is EiB. (This should not be confused with the output control −−units, where capital letters mean multiple of 1000.)

ENVIRONMENT VARIABLES

See **lvm**(8) for information about environment variables used by lvm. For example, LVM_VG_NAME can generally be substituted for a required VG parameter.

EXAMPLES

If the partition table is corrupted or lost on /dev/sda, and you suspect there was an LVM partition at approximately 100 MiB, then this area of the disk can be scanned using the **−−labelsector** parameter with a value of 204800 (100 * 1024 * 1024 / 512 = 204800).

pvck −−labelsector 204800 /dev/sda

SEE ALSO

lvm(8) **lvm.conf**(5) **lvmconfig**(8)

pvchange(8) **pvck**(8) **pvcreate**(8) **pvdisplay**(8) **pvmove**(8) **pvremove**(8) **pvresize**(8) **pvs**(8) **pvscan**(8)

vgcfgbackup(8) **vgcfgrestore**(8) **vgchange**(8) **vgck**(8) **vgcreate**(8) **vgconvert**(8) **vgdisplay**(8) **vgexport**(8) **vgextend**(8) **vgimport**(8) **vgimportclone**(8) **vgmerge**(8) **vgmknodes**(8) **vgreduce**(8) **vgremove**(8) **vgrename**(8) **vgs**(8) **vgscan**(8) **vgsplit**(8)

lvcreate(8) **lvchange**(8) **lvconvert**(8) **lvdisplay**(8) **lvextend**(8) **lvreduce**(8) **lvremove**(8) **lvrename**(8) **lvresize**(8) **lvs**(8) **lvscan**(8)

lvm-fullreport(8) **lvm-lvpoll**(8) **lvm2−activation−generator**(8) **blkdeactivate**(8) **lvmdump**(8)

dmeventd(8) **lvmpolld**(8) **lvmlockd**(8) **lvmlockctl**(8) **cmirrord**(8) **lvmdbusd**(8)

lvmsystemid(7) **lvmreport**(7) **lvmraid**(7) **lvmthin**(7) **lvmcache**(7)