

**NAME**

**thin\_check** - validates thin provisioning metadata on a device or file

**SYNOPSIS**

**thin\_check** [*options*] {device|file}

**DESCRIPTION**

**thin\_check** checks thin provisioning metadata created by the device-mapper thin provisioning target on a device or file.

The tool cannot be run on live metadata unless the **--metadata-snapshot** option is used.

**OPTIONS****-q, --quiet**

Suppress output messages, return only exit code.

**-h, --help**

Print help and exit.

**-V, --version**

Output version information and exit.

**--super-block-only**

Only check the superblock.

**--skip-mappings**

Skip checking of the block mappings which make up the bulk of the metadata.

**--ignore-non-fatal-errors**

Will only return a non-zero exit code if it finds a fatal error.

An example of a nonfatal error is an incorrect data block reference count causing a block to be considered allocated when it in fact isn't. Ignoring errors for a long time is not advised, you really should be using `thin_repair` to fix them.

**--clear-needs-check-flag**

Clears the 'needs\_check' flag in the superblock.

The kernel may set a flag to force the pool to be checked before it's next activated. Set this switch to clear the flag if the check is successful. If the metadata check failed, the flag is not cleared and a `thin_repair` run is needed to fix any issues. After `thin_repair` succeeded, you may run `thin_check` again.

**--metadata-snapshot, -m**

Check the metadata snapshot.

This will check the devices tree and mappings in a metadata snapshot. The snap does not contain space maps, so these will not be checked. This may be used on live metadata.

**--auto-repair**

Automatically repair any trivial issues found with the metadata.

Currently only fixes metadata leaks.

**--override-mapping-root <block>**

Specify a mapping root to use.

Don't use this. This overrides what's specified in the superblock. Only

use this if you really understand the metadata format and are trying to recover damaged metadata.

**EXAMPLE**

Analyses thin provisioning metadata on logical volume /dev/vg/metadata:

```
$ thin_check /dev/vg/metadata
```

The device must not be actively used by the target when running.

**DIAGNOSTICS**

**thin\_check** returns an exit code of 0 for success or 1 for error.

**SEE ALSO**

**thin\_dump(8)**, **thin\_repair(8)**, **thin\_restore(8)**, **thin\_rmap(8)**, **thin\_metadata\_size(8)**

**AUTHOR**

Joe Thornber <ejt@redhat.com>, Heinz Mauelshagen <heinzm@redhat.com>