Rocky Enterprise Linux 9.2 Manual Pages on command 'dmfilemapd.8'

$ man dmfilemapd.8

DMFILEMAPD(8) MAINTENANCE COMMANDS DMFILEMAPD(8)

NAME

dmfilemapd ? device-mapper filemap monitoring daemon

SYNOPSIS

dmfilemapd file_descriptor group_id abs_path inode|path [foreground

[verbose]]

DESCRIPTION

The dmfilemapd daemon monitors groups of dmstats(8) regions that corre?

spond to the extents of a file, adding and removing regions to reflect

the changing state of the file on-disk.

The daemon is normally launched automatically by the dmstats create

command, but can be run manually, either to create a new daemon where

one did not previously exist, or to change the options previously used,

by killing the existing daemon and starting a new one.

OPTIONS

file_descriptor

Specify the file descriptor number for the file to be monitored.

The file descriptor must reference a regular file, open for

reading, in a local file system that supports the FIEMAP ioctl, and that returns data describing the physical location of ex? tents.

The process that executes dmfilemapd is responsible for opening the file descriptor that is handed to the daemon.

group_id

The dmstats group identifier of the group that dmfilemapd should update. The group must exist and it should correspond to a set of regions created by a previous filemap operation.

abs_path

The absolute path to the file being monitored, at the time that it was opened. The use of abs_path by the daemon differs, de? pending on the filemap following mode in use; see MODES and the mode option for more information.

inode|path

The filemap monitoring mode the daemon. Use either inode (DM_FILEMAP_FOLLOW_INODE), or path (DM_FILEMAP_FOLLOW_PATH), to enable follow-inode or follow-path mode respectively.

[foreground]

If set to 1, disable forking and allow the daemon to run in the foreground.

[verbose]

Control daemon logging. If set to zero, the daemon will close all stdio streams and run silently. If verbose is a number be? tween 1 and 3, stdio will be retained and the daemon will log messages to stdout and stderr that match the specified verbosity level.

MODES

The file map monitoring daemon can monitor files in two distinct ways: the mode affects the behaviour of the daemon when a file under monitor? ing is renamed or unlinked, and the conditions which cause the daemon to terminate.

In both modes, the daemon will always shut down when the group being

monitored is deleted.

Follow inode

The daemon follows the inode of the file, as it was at the time the daemon started. The file descriptor referencing the file is kept open at all times, and the daemon will exit when it detects that the file has been unlinked and it is the last holder of a reference to the file. This mode is useful if the file is expected to be renamed, or moved within the file system, while it is being monitored.

Follow path

The daemon follows the path that was given on the daemon command line. The file descriptor referencing the file is re-opened on each iteration of the daemon, and the daemon will exit if no file exists at this loca? tion (a tolerance is allowed so that a brief delay between removal and replacement is permitted).

This mode is useful if the file is updated by unlinking the original and placing a new file at the same path.

LIMITATIONS

The daemon attempts to maintain good synchronisation between the file extents and the regions contained in the group, however, since the dae? mon can only react to new allocations once they have been written, there are inevitably some IO events that cannot be counted when a file is growing, particularly if the file is being extended by a single thread writing beyond EOF (for example, the dd program).

There is a further loss of events in that there is currently no way to atomically resize a dmstats region and preserve its current counter values. This affects files when they grow by extending the final ex? tent, rather than allocating a new extent: any events that had accumu? lated in the region between any prior operation and the resize are lost.

File mapping is currently most effective in cases where the majority of IO does not trigger extent allocation. Future updates may address these limitations when kernel support is available.

EXAMPLES

Normally the daemon is started automatically by the dmstats create or update_filemap commands but it can be run manually for debugging or testing purposes.

Start the daemon in the background, in follow-path mode

dmfilemapd 3 0 /srv/images/vm.img path 0 0 3< /srv/images/vm.img

Start the daemon in follow-inode mode, disable forking and enable ver? bose logging

dmfilemapd 3 0 /var/tmp/data inode 1 3 3< /var/tmp/data

Starting dmfilemapd with fd=3, group_id=0 mode=inode, path=/var/tmp/data

dm version [opencount flush] [16384] (*1)

dm info (253:0) [opencount flush] [16384] (*1)

dm message (253:0) [opencount flush] @stats_list dmstats [16384] (*1)

Read alias 'data' from aux_data

Found group_id 0: alias="data"

dm_stats_walk_init: initialised flags to 4000000000000

starting stats walk with GROUP

exiting _filemap_monitor_get_events() with deleted=0, check=0

Waiting for check interval

AUTHORS

Bryn M. Reeves <bmr@redhat.com>

SEE ALSO

dmstats(8)

LVM2 resource page: ?https://www.sourceware.org/lvm2?

Device-mapper resource page: ?http://sources.redhat.com/dm?