Rocky Enterprise Linux 9.2 Manual Pages on command 'systemd.path.5'

C:\>man systemd.path.5

SYSTEMD.PATH(5) systemd.path SYSTEMD.PATH(5)

NAME

systemd.path - Path unit configuration

SYNOPSIS

path.path

DESCRIPTION

A unit configuration file whose name ends in ".path" encodes information about a path monitored by systemd, for path-based activation.

This man page lists the configuration options specific to this unit type. See systemd.unit(5) for the common options of all unit configuration files. The common configuration items are configured in the generic "[Unit]" and "[Install]" sections. The path specific configuration options are configured in the "[Path]" section.

For each path file, a matching unit file must exist, describing the unit to activate when the path changes. By default, a service by the same name as the path (except for the suffix) is activated. Example: a path file foo.path activates a matching service foo.service. The unit to activate may be controlled by Unit= (see below).

Internally, path units use the inotify(7) API to monitor file systems. Due to that, it suffers by the same limitations as inotify, and for example cannot be used to monitor files or directories changed by other machines on remote NFS file systems.

When a service unit triggered by a path unit terminates (regardless whether it

exited successfully or failed), monitored paths are checked immediately again, and the service accordingly restarted instantly. As protection against busy looping in this trigger/start cycle, a start rate limit is enforced on the service unit, see StartLimitIntervalSec= and StartLimitBurst= in systemd.unit(5). Unlike other service failures, the error condition that the start rate limit is hit is propagated from the service unit to the path unit and causes the path unit to fail as well, thus ending the loop.

AUTOMATIC DEPENDENCIES

Implicit Dependencies

The following dependencies are implicitly added:

- If a path unit is beneath another mount unit in the file system hierarchy, both a requirement and an ordering dependency between both units are created automatically.
- An implicit Before= dependency is added between a path unit and the unit it is supposed to activate.

Default Dependencies

The following dependencies are added unless DefaultDependencies=no is set:

- Path units will automatically have dependencies of type Before= on paths.target, dependencies of type After= and Requires= on sysinit.target, and have dependencies of type Conflicts= and Before= on shutdown.target. These ensure that path units are terminated cleanly prior to system shutdown. Only path units involved with early boot or late system shutdown should disable DefaultDependencies= option.

OPTIONS

Path files must include a [Path] section, which carries information about the path(s) it monitors. The options specific to the [Path] section of path units are the following:

PathExists=, PathExistsGlob=, PathChanged=, PathModified=, DirectoryNotEmpty=

Defines paths to monitor for certain changes: PathExists= may be used to watch the mere existence of a file or directory. If the file specified exists, the configured unit is activated. PathExistsGlob= works similar, but checks for the existence of at least one file matching the globbing pattern specified. PathChanged= may be used to watch a file or directory and activate the

configured unit whenever it changes. It is not activated on every write to the watched file but it is activated if the file which was open for writing gets closed. PathModified= is similar, but additionally it is activated also on simple writes to the watched file. DirectoryNotEmpty= may be used to watch a directory and activate the configured unit whenever it contains at least one file.

The arguments of these directives must be absolute file system paths.

Multiple directives may be combined, of the same and of different types, to watch multiple paths. If the empty string is assigned to any of these options, the list of paths to watch is reset, and any prior assignments of these options will not have any effect.

If a path already exists (in case of PathExists= and PathExistsGlob=) or a directory already is not empty (in case of DirectoryNotEmpty=) at the time the path unit is activated, then the configured unit is immediately activated as well. Something similar does not apply to PathChanged= and PathModified=.

If the path itself or any of the containing directories are not accessible, systemd will watch for permission changes and notice that conditions are satisfied when permissions allow that.

Unit=

The unit to activate when any of the configured paths changes. The argument is a unit name, whose suffix is not ".path". If not specified, this value defaults to a service that has the same name as the path unit, except for the suffix. (See above.) It is recommended that the unit name that is activated and the unit name of the path unit are named identical, except for the suffix.

MakeDirectory=

Takes a boolean argument. If true, the directories to watch are created before watching. This option is ignored for PathExists= settings. Defaults to false.

DirectoryMode=

If MakeDirectory= is enabled, use the mode specified here to create the directories in question. Takes an access mode in octal notation. Defaults to 0755.

SEE ALSO

systemd(1), systemctl(1), systemd.unit(5), systemd.service(5), inotify(7),