Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'systemd-udevd.service.8' command

$ man systemd-udevd.service.8

SYSTEMD-UDEVD.SERVICE(8) systemd-udevd.service SYSTEMD-UDEVD.SERVICE(8)

NAME

systemd-udevd.service, systemd-udevd-control.socket, systemd-udevd-kernel.socket, systemd-udevd - Device event managing daemon

SYNOPSIS

systemd-udevd.service

systemd-udevd-control.socket

systemd-udevd-kernel.socket

/usr/lib/systemd/systemd-udevd [--daemon] [--debug] [--children-max=]
[--exec-delay=] [--event-timeout=]
[--resolve-names=early|late|never]
[--version] [--help]

DESCRIPTION

systemd-udevd listens to kernel uevents. For every event, systemd-udevd executes matching instructions specified in udev rules. See udev(7).

The behavior of the daemon can be configured using udev.conf(5), its command line options, environment variables, and on the kernel command line, or changed dynamically with udevadm control.

OPTIONS

-d, --daemon

Detach and run in the background.

-D, --debug

Print debug messages to standard error.

-c, --children-max=

Limit the number of events executed in parallel.

-e, --exec-delay=

Delay the execution of each RUN{program} parameter by the given number of seconds. This option might be useful when debugging system crashes during coldplug caused by loading non-working kernel modules.

-t, --event-timeout=

Set the number of seconds to wait for events to finish. After this time, the event will be terminated. The default is 180 seconds.

-s, --timeout-signal=

Set the signal which systemd-udevd will send to forked off processes after reaching event timeout. The setting can be overridden at boot time with the kernel command line option udev.timeout_signal=. Setting to SIGABRT may be helpful in order to debug worker timeouts. Defaults to SIGKILL. Note that setting the option on the command line overrides the setting from the configuration file.

-N, --resolve-names=

Specify when systemd-udevd should resolve names of users and groups. When set to early (the default), names will be resolved when the rules are parsed. When set to late, names will be resolved for every event. When set to never, names will never be resolved and all devices will be owned by root.

-h, --help

Print a short help text and exit.

--version

Print a short version string and exit.

KERNEL COMMAND LINE

Parameters prefixed with "rd." will be read when systemd-udevd is used in an initrd, those without will be processed both in the initrd and on the host.

udev.log_level=, rd.udev.log_level=

Set the log level.

udev.children_max=, rd.udev.children_max=

Limit the number of events executed in parallel.

udev.exec_delay=, rd.udev.exec_delay=

Delay the execution of each RUN{program} parameter by the given number of seconds. This option might be useful when debugging system crashes during coldplug caused by loading non-working kernel modules.

udev.event_timeout=, rd.udev.event_timeout=

Wait for events to finish up to the given number of seconds. This option might be useful if events are terminated due to kernel drivers taking too long to initialize.

udev.timeout_signal=, rd.udev.timeout_signal=

Specifies a signal that systemd-udevd will send to workers on timeout. Note that kernel command line option overrides both the setting in the configuration file and the one on the program command line.

udev.blockdev_read_only, rd.udev.blockdev_read_only

If specified, mark all physical block devices read-only as they appear. Synthetic block devices (such as loopback block devices or device mapper devices) are left as they are. This is useful to guarantee that the contents of physical block devices remains unmodified during runtime, for example to implement fully stateless systems, for testing or for recovery situations where corrupted file systems shall not be corrupted further through accidental modification.

A block device may be marked writable again by issuing the blockdev --setrw command, see blockdev(8) for details.

net.ifnames=

Network interfaces are renamed to give them predictable names when possible. It is enabled by default; specifying 0 disables it.

net.naming-scheme=

Network interfaces are renamed to give them predictable names when

possible (unless net.ifnames=0 is specified, see above). With this kernel command line option it is possible to pick a specific version of this algorithm and override the default chosen at compilation time. Expects one of the naming scheme identifiers listed in systemd.net-naming-scheme(7), or "latest" to select the latest scheme known (to this particular version of systemd-udevd.service).

Note that selecting a specific scheme is not sufficient to fully stabilize interface naming: the naming is generally derived from driver attributes exposed by the kernel. As the kernel is updated, previously missing attributes systemd-udevd.service is checking might appear, which affects older name derivation algorithms, too.

net.ifname-policy=policy1[,policy2,...][,MAC]

Specifies naming policies applied when renaming network interfaces. Takes a list of policies and an optional MAC address separated with comma. Each policy value must be one of the policies understood by the NamePolicy= setting in .link files, e.g. "onboard" or "path". See systemd.link(5) for more details. When the MAC address is specified, the policies are applied to the interface which has the address. When no MAC address is specified, the policies are applied to all interfaces. This kernel command line argument can be specified multiple times.

This argument is not directly read by systemd-udevd, but is instead converted to a .link file by systemd-network-generator.service(8). For this argument to take effect, systemd-network-generator.service must be enabled.

Example:

```
net.ifname-policy=keep,kernel,path,slot,onboard,01:23:45:67:89:ab
net.ifname-policy=keep,kernel,path,slot,onboard,mac
```

This is mostly equivalent to creating the following .link files:

```
# 91-name-policy-with-mac.link
[Match]
MACAddress=01:23:45:67:89:ab
```

[Link]

NamePolicy=keep kernel path slot onboard

AlternativeNamePolicy=path slot onboard

and

92-name-policy-for-all.link

[Match]

OriginalName=*

[Link]

NamePolicy=keep kernel path slot onboard mac

AlternativeNamePolicy=path slot onboard mac

SEE ALSO

udev.conf(5), udev(7), udevadm(8)