



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

Rocky Enterprise Linux 9.2 Manual Pages on command 'systemd-socket-activate.1'

\$ man systemd-socket-activate.1

SYSTEMD-SOCKET-ACTIVATE(1) systemd-socket-activate SYSTEMD-SOCKET-ACTIVATE(1)

NAME

systemd-socket-activate - Test socket activation of daemons

SYNOPSIS

systemd-socket-activate [OPTIONS...] daemon [OPTIONS...]

DESCRIPTION

systemd-socket-activate may be used to launch a socket-activated service program from the command line for testing purposes. It may also be used to launch individual instances of the service program per connection.

The daemon to launch and its options should be specified after options intended for systemd-socket-activate.

If the --inetd option is given, the socket file descriptor will be used as the standard input and output of the launched process. Otherwise, standard input and output will be inherited, and sockets will be passed through file descriptors 3 and higher. Sockets passed through \$LISTEN_FDS to systemd-socket-activate will be passed through to the daemon, in the original positions. Other sockets specified with

--listen= will use consecutive descriptors. By default, systemd-socket-activate listens on a stream socket, use --datagram and --seqpacket to listen on datagram or sequential packet sockets instead (see below).

OPTIONS

-l address, --listen=address

Listen on this address. Takes a string like "2000" or "127.0.0.1:2001".

-a, --accept

Launch an instance of the service program for each connection and pass the connection socket.

-d, --datagram

Listen on a datagram socket (SOCK_DGRAM), instead of a stream socket (SOCK_STREAM). May not be combined with --seqpacket.

--seqpacket

Listen on a sequential packet socket (SOCK_SEQPACKET), instead of a stream socket (SOCK_STREAM). May not be combined with --datagram.

--inetd

Use the inetd protocol for passing file descriptors, i.e. as standard input and standard output, instead of the new-style protocol for passing file descriptors using \$LISTEN_FDS (see above).

-E VAR[=VALUE], --setenv=VAR[=VALUE]

Add this variable to the environment of the launched process. If VAR is followed by "=", assume that it is a variable?value pair.

Otherwise, obtain the value from the environment of systemd-socket-activate itself.

--fdname=NAME[:NAME...]

Specify names for the file descriptors passed. This is equivalent to setting FileDescriptorName= in socket unit files, and enables use of sd_listen_fds_with_names(3). Multiple entries may be specified using separate options or by separating names with colons (":") in one option. In case more names are given than descriptors,

superfluous ones will be ignored. In case less names are given than descriptors, the remaining file descriptors will be unnamed.

`-h, --help`

Print a short help text and exit.

`--version`

Print a short version string and exit.

ENVIRONMENT VARIABLES

`$LISTEN_FDS`, `$LISTEN_PID`, `$LISTEN_FDNames`

See `sd_listen_fds(3)`.

`$SYSTEMD_LOG_TARGET`, `$SYSTEMD_LOG_LEVEL`, `$SYSTEMD_LOG_TIME`,

`$SYSTEMD_LOG_COLOR`, `$SYSTEMD_LOG_LOCATION`

Same as in `systemd(1)`.

EXAMPLES

Example 1. Run an echo server on port 2000

```
$ systemd-socket-activate -l 2000 --inetd -a cat
```

Example 2. Run a socket-activated instance of `systemd-journal-gatewayd(8)`

```
$ systemd-socket-activate -l 19531 /usr/lib/systemd/systemd-journal-gatewayd
```

SEE ALSO

`systemd(1)`, `systemd.socket(5)`, `systemd.service(5)`, `systemd-run(1)`,

`sd_listen_fds(3)`, `sd_listen_fds_with_names(3)`, `cat(1)`

`systemd` 252

`SYSTEMD-SOCKET-ACTIVATE(1)`