NAME

pppctl - PPP control program

SYNOPSIS

pppctl [**-v**] [**-t** *n*] [**-p** *passwd*] [*host*:]*Port* | *LocalSocket* [*command*[;*command*]...]

DESCRIPTION

This utility provides command line control of the ppp(8) daemon. Its primary use is to facilitate simple scripts that control a running daemon.

The **pppctl** utility is passed at least one argument, specifying the socket on which **ppp** is listening. Refer to the 'set server' command of **ppp** for details. If the socket contains a leading '/', it is taken as an AF_LOCAL socket. If it contains a colon, it is treated as a *host*:*port* pair, otherwise it is treated as a TCP port specification on the local machine (127.0.0.1). Both the *host* and *port* may be specified numerically if you wish to avoid a DNS lookup or do not have an entry for the given port in */etc/services*.

All remaining arguments are concatenated to form the *command*(s) that will be sent to the **ppp** daemon. If any semi-colon characters are found, they are treated as *command* delimiters, allowing more than one *command* in a given 'session'. For example:

    pppctl 3000 set timeout 300\; show timeout

Do not forget to escape or quote the ';' as it is a special character for most shells.

If no *command* arguments are given, **pppctl** enters interactive mode, where commands are read from standard input. When reading commands, the editline(3) library is used, allowing command-line editing (with editrc(5) defining editing behaviour). The history size defaults to *20 lines*.

The following command line options are available:

-v   Display all data sent to and received from the **ppp** daemon. Normally, **pppctl** displays only non-prompt lines received. This option is ignored in interactive mode.

-t *n*   Use a timeout of *n* instead of the default 2 seconds when connecting. This may be required if you wish to control a daemon over a slow (or even a dialup) link.

-p *passwd*
        Specify the password required by the **ppp** daemon. If this switch is not used, **pppctl** will prompt for a password once it has successfully connected to **ppp**.

ENVIRONMENT

The following environment variables are understood by **pppctl** when in interactive mode:

EL_SIZE   The number of history lines. The default is 20.

EL_EDITOR   The edit mode. Only values of "emacs" and "vi" are accepted. Other values are
                 silently ignored. This environment variable will override the *bind -v* and *bind -e*
                 commands in *~/.editrc*.

EXAMPLES

If you run **ppp** in **-auto** mode, **pppctl** can be used to automate many frequent tasks (you can actually
control **ppp** in any mode except interactive mode). Use of the **-p** option is discouraged (even in scripts
that are not readable by others) as a ps(1) listing may reveal your secret.

The best way to allow easy, secure **pppctl** access is to create a local server socket in *etc/ppp/ppp.conf*
(in the correct section) like this:

    set server /var/run/internet "" 0177

This will instruct **ppp** to create a local domain socket, with srw------- permissions and no password,
allowing access only to the user that invoked **ppp**. Refer to the ppp(8) man page for further details.

You can now create some easy-access scripts. To connect to the internet:

    #! /bin/sh
    test $# -eq 0 && time=300 || time=$1
    exec pppctl /var/run/internet set timeout $time\; dial

To disconnect:

    #! /bin/sh
    exec pppctl /var/run/internet set timeout 300\; close

To check if the line is up:

    #! /bin/sh
    pppctl -p '' -v /var/run/internet quit | grep ^PPP >/dev/null
    if [$? -eq 0]; then
      echo Link is up
    else

```
    echo Link is down
fi
```

You can even make a generic script:

```
#! /bin/sh
exec pppctl /var/run/internet "$@"
```

You could also use **pppctl** to control when dial-on-demand works. Suppose you want **ppp** to run all the time, but you want to prevent dial-out between 8pm and 8am each day. However, any connections active at 8pm should continue to remain active until they are closed or naturally time out.

A cron(8) entry for 8pm which runs

```
pppctl /var/run/internet set filter dial 0 deny 0 0
```

will block all further dial requests, and the corresponding 8am entry

```
pppctl /var/run/internet set filter dial -1
```

will allow them again.

SEE ALSO
  ps(1), editline(3), editrc(5), services(5), ppp(8)

HISTORY
  The **pppctl** utility first appeared in FreeBSD 2.2.5.