

**NAME**

**pon**, **poff**, **plog** – starts up, shuts down or lists the log of PPP connections

**SYNOPSIS**

```
pon [ isp-name [ options ] ]
poff [ -r ] [ -d ] [ -c ] [ -a ] [ -h ] [ isp-name ]
plog [ arguments ]
```

**DESCRIPTION**

This manual page describes the **pon**, **plog** and **poff** scripts, which allow users to control PPP connections.

**pon**

**pon**, invoked without arguments, runs the */etc/ppp/ppp\_on\_boot* file, if it exists and is executable. Otherwise, a PPP connection will be started using configuration from */etc/ppp/peers/provider*. This is the default behaviour unless an **isp-name** argument is given.

For instance, to use ISP configuration "myisp" run:

```
pon myisp
```

**pon** will then use the options file */etc/ppp/peers/myisp*. You can pass additional **pppd options** after the ISP name, too. **pon** can be used to run multiple, simultaneous PPP connections.

**pon** takes the following command line options:

```
-q --quick
    disconnect when ip-up finishes running. This function is only available to the root user.
```

**poff**

**poff** closes a PPP connection. If more than one PPP connection exists, the one named in the argument to **poff** will be killed, e.g.

```
poff myprovider2
```

will terminate the connection to myprovider2, and leave the PPP connections to e.g. "myprovider1" or "myprovider3" up and running.

**poff** takes the following command line options:

```
-r      causes the connection to be redialed after it is dropped.
-d      toggles the state of pppd's debug option.
-c      causes pppd(8) to renegotiate compression.
-a      stops all running ppp connections. If the argument isp-name is given it will be ignored.
-h      displays help information.
-v      prints the version and exits.
```

If no argument is given, **poff** will stop or signal pppd if and only if there is exactly one running. If more than one connection is active, it will exit with an error code of 1.

**plog**

**plog** shows you the last few lines of */var/log/ppp.log*. If that file doesn't exist, it shows you the last few lines of your */var/log/syslog* file, but excluding the lines not generated by pppd. This script makes use of the **tail**(1) command, so arguments that can be passed to **tail**(1) can also be passed to **plog**.

Note: the **plog** script can only be used by root or another system administrator in group "adm", due to security reasons. Also, to have all pppd-generated information in one logfile, that plog can show, you need the following line in your */etc/syslog.conf* file:

```
local2.*      -/var/log/ppp.log
```

**FILES**

*/etc/ppp/options*

PPPd system options file.

*/etc/ppp/pap-secrets*

System PAP passwords file.

*/etc/ppp/chap-secrets*

System CHAP passwords file.

*/etc/ppp/peers/*

Directory holding the peer options files. The default file is called *provider*.

*/etc/chatscripts/provider*

The chat script invoked from the default */etc/ppp/peers/provider*.

*/var/log/ppp.log*

The default PPP log file.

## AUTHORS

The p-commands were written by Christoph Lameter <clameter@debian.org>. Updated and revised by Philip Hands <phil@hands.com>.

This manual was written by Othmar Pasteka <othmar@tron.at>. Modified by Rob Levin <lilo@openprojects.net>, with some extensions taken from the old p-commands manual written by John Hasler <jhasler@debian.org>.

## SEE ALSO

**pppd(8)**, **chat(8)**, **tail(1)**.