

pure-ftpd(8) - Linux man page

Name

pure-ftpd - simple File Transfer Protocol server

Synopsis

**pure-ftpd [-0] [-1] [-4] [-6] [-a gid] [-A] [-b] [-B] [-c clients] [-C cnx/ip] [-d
[-d]] [-D] [-e] [-E] [-f facility] [-F fortunes file] [-g pidfile] [-G] [-H] [-i] [-I]
[-j] [-k percentage] [-K] [-l authentication[:config file]] [-L max files:max
depth] [-m maxload] [-M] [-n maxfiles:maxsize] [-N] [-o] [-O format:log file]
[-p first:last] [-P ip address or host name] [-q upload:download ratio] [-Q
upload:download ratio] [-r] [-R] [-s] [-S [address,][port]] [-t upload
bandwidth:download bandwidth] [-T upload bandwidth:download bandwidth]
[-u uid] [-U umask files:umask dirs] [-v bonjour name] [-V ip address] [-w]
[-W] [-x] [-X] [-y max user sessions:max anon sessions] [-Y tls behavior] [-z]
[-Z]**

Alternative style :

-0 --nottruncate
-1 --logpid
-4 --ipv4only
-6 --ipv6only
-a --trustedgid
-A --chrooteveryone
-b --brokenclientscompatibility
-B --daemonize
-c --maxclientsnumber
-C --maxclientsperip
-d --verboselog
-D --displaydotfiles
-e --anonymousonly
-E --noanonymous
-f --syslogfacility
-F --fortunesfile
-g --pidfile
-G --norename
-h --help
-H --dontresolve
-i --anonymouscantupload
-I --maxidletime
-j --createhomedir

- k --maxdiskusagepct
- K --keepallfiles
- l --login
- L --limitrecursion
- m --maxload
- M --anonymouscancreatedirs
- n --quota
- N --natmode
- o --uploadscript
- O --altlog
- p --passiveportrange
- P --forcepassiveip
- q --anonymoustratio
- Q --userratio
- r --autorename
- R --nochmod
- s --antiwarez
- S --bind
- t --anonymousbandwidth
- T --userbandwidth
- u --minuid
- U --umask
- v --bonjour
- V --trustedip
- w --allowuserfxp
- W --allowanonymousfxp
- x --prohibitdotfileswrite
- X --prohibitdotfilesread
- y --peruserlimits
- Y --tls
- z --allowdotfiles
- Z --customerproof

Description

Pure-FTPd is a small, simple server for the old and hairy File Transfer Protocol, designed to use less resources than older servers, be smaller and very secure, and to never execute any external program.

It support most-used features and commands of FTP (including many modern extensions), and leaves out everything which is deprecated, meaningless, insecure, or correlates with trouble.

IPv6 is fully supported.

Options

-0

When a file is uploaded and there is already a previous version of the file with the same name, the old file will neither get removed nor truncated. Upload will take place in a temporary file and once the upload is complete, the switch to the new version will be atomic. This option should not be used together with virtual quotas.

-1

Add the PID to the syslog output. Ignored if **-f none** is set.

-4

Listen only to IPv4 connections.

-6

Listen only to IPv6 connections.

-a

gid Regular users will be chrooted to their home directories, unless they belong to the specified gid. Note that root is always trusted, and that chroot() occurs only for anonymous ftp without this option.

-A Chroot() everyone, but root.

-b Be broken. Turns on some compatibility hacks for shoddy clients, and for broken Netfilter gateways.

-B Start the standalone server in background (daemonize).

-c clients

Allow a maximum of *clients* to be connected. *clients* must be at least 1, and if you combine it with **-p** it will be forced down to half the number of ports specified by **-p**. If more than *clients* are connected, new clients are rejected at once, even clients wishing to upload, or to log in as normal users. Therefore, it is advisable to use **-m** as primary overload protection. The default value is 50.

-C max connection per ip

Limit the number of simultaneous connections coming from the same IP address.

This is yet another very effective way to prevent stupid denial of services and bandwidth starvation by a single user. It works only when the server is launched in standalone mode (if you use a super-server, it is supposed to do that). If the server is launched with **-C 2**, it doesn't mean that the total number of connection is limited to 2. But the same client, coming from the same machine (or at least the

same IP), can't have more than two simultaneous connections. This features needs some memory to track IP addresses, but it's recommended to use it.

-d

turns on debug logging. Every command is logged, except that the argument to PASS is changed to "<password>". If you repeat **-d** , responses too are logged.

-e

Only allow anonymous users to log in.

-E

Only allow authenticated login. Anonymous users are prohibited.

-f facility

makes ftpd use *facility* for all **syslog**(3) messages. *facility* defaults to **ftp**. The facility names are normally listed in [/usr/include/sys/syslog.h](#). Note that if **-f** is not the first option on the command line, a couple of messages may be logged to local2 before the **-f** option is parsed. Use **-f none** to disable logging.

-F fortunes file

Display a funny random message in the initial login banner. The random cookies are extracted from a text file, in the standard **fortune** format. If you installed the **fortune** package, you should have a directory (usually **/usr/share/fortune**) with binary files (**xxxx.dat**) and text files (without the **.dat** extension).

-g pidfile

In standalone mode, write the pid to that file in instead of /var/run/pure-ftpd.pid .

-G

When this option is enabled, people can no more change the name of already uploaded files, even if they own those files or their directory.

-H

Don't resolve host names ("192.0.34.166" will be logged instead of "www.example.com"). It can significantly speed up connections and reduce bandwidth usage on busy servers. Use it especially on public FTP sites.

-i

Disallow upload for anonymous users, whatever directory permissions are. This option is especially useful for virtual hosting, to avoid your users create warez sites in their account.

-I timeout

Change the maximum idle time. The timeout is in minutes, and defaults to 15.

-j

If the home directory of a user doesn't exist, automatically create it. The newly created home directory belongs to the user, and permissions are set according to the current directory mask. To avoid local attacks, the parent directory should never belong to an untrusted user.

-k percentage

Disallow upload if the partition is more than **percentage** full. Example: -k 95 will ensure that your disk will never get filled more than 95% by FTP users.

-K

Allow users to resume and upload files, but NOT to delete them. Directories can be removed, but only if they are empty.

-l authentication:file

Enable a new authentication method. It can be one of : *-l unix* For standard (/etc/passwd) authentication. *-l pam* For PAM authentication. *-l ldap:LDAP config file* For LDAP directories. *-l mysql:MySQL config file* For MySQL databases. *-l postgres:Postgres config file* For Postgres databases. *-l puredb:PureDB database file* For PureDB databases. *-l extauth:path to pure-authd socket* For external authentication handlers.

Different authentication methods can be mixed together. For instance if you run the server with *-lpuredb:/etc/pure-ftpd/pwd.pdb -lmysql:/etc/pure-ftpd/my.cf -lunix* Accounts will first be authenticated from a PureDB database. If it fails, a MySQL server will be asked. If the account is still not found in the database, standard unix accounts will be scanned. Authentication methods are tried in the order you give the -l options.

See the *README.LDAP* and *README.MySQL* files for info about the built-in LDAP and SQL directory support.

-L max files:max depth

Avoid denial-of-service attacks by limiting the number of displayed files in a 'ls' and the maximum depth of a recursive 'ls'. Defaults are 2000:5 (2000 files displayed for a single 'ls' and walk through 5 subdirectories max).

-m load

Do not allow anonymous users to download files if the load is above *load* when the user connects. Uploads and file listings are still allowed, as are downloads by real users. The user is not told about this until he/she tries to download a file.

-M

Allow anonymous users to create directories.

-n maxfiles:maxsize

Enable **virtual quotas** When virtual quotas are enabled, .ftpquota files are created,

and the number of files for a user is restricted to 'maxfiles'. The max total size of his directory is also restricted to 'maxsize' Megabytes. Members of the trusted group aren't subject to quotas.

-N

NAT mode. Force **active** mode. If your FTP server is behind a NAT box that doesn't support applicative FTP proxying, or if you use port redirection without a transparent FTP proxy, use this. Well... the previous sentence isn't very clear. Okay: if your network looks like this:

FTP--NAT.gateway/router--Internet

and if you want people coming from the internet to have access to your FTP server, please try without this option first. If Netscape clients can connect without any problem, your NAT gateway rulez. If Netscape doesn't display directory listings, your NAT gateway sucks. Use **-N** as a workaround.

-O

Enable *pure-uploadscript*.

-O format:log file

Record all file transfers into a specific log file, in an alternative format. Currently, three formats are supported : CLF, Stats, W3C and xferlog.

If you add

-O clf:/var/log/pureftpd.log

to your starting options, Pure-FTPd will log transfers in **/var/log/pureftpd.log** in a format similar to the Apache web server in default configuration.

If you add

-O stats:/var/log/pureftpd.log

to your starting options, Pure-FTPd will create accurate log files designed for traffic analys software like ftpStats.

If you add

-O w3c:/var/log/pureftpd.log

to your starting options, Pure-FTPd will create W3C-conformant log files.

For security purposes, the path must be absolute (eg. **/var/log/pureftpd.log**, not **../log/pureftpd.log**).

-p first:last

Use only ports in the range *first* to *last* inclusive for passive-mode downloads. This means that clients will not try to open connections to TCP ports outside the range *first* - *last*, which makes pure-ftpd more compatible with packet filters. Note that the maximum number of clients (specified with **-c**) is forced down to $(last + 1 - first)/2$ if it is greater, as the default is. (The syntax for the port range is, conveniently, the same as that of **iptables**).

-P ip address or host name

Force the specified IP address in reply to a **PASV/EPSV/SPSV** command. If the

server is behind a masquerading (NAT) box that doesn't properly handle stateful FTP masquerading, put the ip address of that box here. If you have a dynamic IP address, you can use a symbolic host name (probably the one of your gateway), that will be resolved every time a new client will connect.

-q upload:download

Enable an upload/download ratio for anonymous users (ex: `-q 1:5` means that 1 Mb of goodies have to be uploaded to leech 5 Mb).

-Q upload:download

Enable ratios for anonymous and non-anonymous users. If the `-a` option is also used, users from the trusted group have no ratio.

-r

Never overwrite existing files. Uploading a file whose name already exists cause an automatic rename. Files are called `xyz.1`, `xyz.2`, `xyz.3`, etc.

-R

Disallow users (even non-anonymous ones) usage of the `CHMOD` command. On hosting services, it may prevent newbies from doing mistakes, like setting bad permissions on their home directory. Only root can use `CHMOD` when this switch is enabled.

-s

Don't allow anonymous users to retrieve files owned by "ftp" (generally, files uploaded by other anonymous users).

-S [{ip address|hostname}] [, {port|service name}]

This option is only effective when the server is launched as a standalone server. Connections are accepted on the specified IP and port. IPv4 and IPv6 are supported. Numeric and fully-qualified host names are accepted. A service name (see `/etc/services`) can be used instead of a numeric port number.

-t bandwidth

or **-t upload bandwidth:download bandwidth** Enable process priority lowering and bandwidth throttling for anonymous users. Delay should be in kilobytes/seconds.

-T bandwidth

or **-T upload bandwidth:download bandwidth** Enable process priority lowering and bandwidth throttling for `*ALL*` users. Pure-FTPd should have been explicitly compiled with throttling support to have these flags work. It is possible to have different bandwidth limits for uploads and for downloads. `-t` and `-T` can indeed be followed by two numbers delimited by a column (':'). The first number is the upload bandwidth and the next one applies only to downloads. One of them can be left blank which means infinity. A single number without any column means that the same limit applies to upload and download.

-u uid

Do not allow uids below *uid* to log in (typically, low-numbered *uids* are used for administrative accounts). **-u 100** is sufficient to deny access to all administrative accounts on many linux boxes, where 99 is the last administrative account. Anonymous FTP is allowed even if the uid of the ftp user is smaller than *uid*. **-u 1** denies access only to root accounts. The default is to allow FTP access to all accounts.

-U umask files:umask dirs

Change the mask for creation of new files and directories. The default are 133 (files are readable -but not writable- by other users) and 022 (same thing for directory, with the execute bit on). If new files should only be readable by the user, use 177:077. If you want uploaded files to be executable, use 022:022 (files will be readable by other people) or 077:077 (files will only be readable by their owner).

-v bonjour name

Set the Bonjour name of the service (only available on MacOS X when Bonjour support is compiled in).

-V ip address

Allow non-anonymous FTP access only on this specific local IP address. All other IP addresses are only anonymous. With that option, you can have routed IPs for public access, and a local IP (like 10.x.x.x) for administration. You can also have a routable trusted IP protected by firewall rules, and only that IP can be used to login as a non-anonymous user.

-w

Enable support for the FXP protocol, for non-anonymous users only.

-W

Enable the FXP protocol for everyone. *FXP IS AN UNSECURE PROTOCOL. NEVER ENABLE IT ON UNTRUSTED NETWORKS.*

-x

In normal operation mode, authenticated users can read/write files beginning with a dot ('.'). Anonymous users can't, for security reasons (like changing banners or a forgotten .rhosts). When '-x' is used, authenticated users can download dot-files, but not overwrite/create them, even if they own them. That way, you can prevent hosted users from messing .qmail files.

-X

This flag is identical to the previous one (writing dot-files is prohibited), but in addition, users can't even **read** files and directories beginning with a dot (like "cd .ssh").

-y per user max sessions:max anonymous sessions

This switch enables per-user concurrency limits. Two values are separated by a column. The first one is the max number of concurrent sessions for a single login. The second one is the maximum number of anonymous sessions.

-Y tls behavior

-Y 0 (default) disables SSL/TLS security mechanisms.

-Y 1 Accept both normal sessions and SSL/TLS ones.

-Y 2 refuses connections that aren't using SSL/TLS security mechanisms, including anonymous ones.

-Y 3 refuses connections that aren't using SSL/TLS security mechanisms, and refuse cleartext data channels as well.

The server must have been compiled with SSL/TLS support and a valid certificate must be in place to accept encrypted sessions.

-z

Allow anonymous users to read files and directories starting with a dot ('.').

-Z

Add safe guards against common customer mistakes (like chmod 0 on their own files) .

Authentication

Some of the complexities of older servers are left out.

This version of pure-ftpd can use PAM for authentication. If you want it to consult any files like /etc/shells or /etc/ftpd/ftpusers consult pam docs. LDAP directories and SQL databases are also supported.

Anonymous users are authenticated in any of three ways:

1. The user logs in as "ftp" or "anonymous" and there is an account called "ftp" with an existing home directory. This server does not ask anonymous users for an email address or other password.
2. The user connects to an IP address which resolves to the name of a directory in */etc/pure-ftpd/pure-ftpd* (or a symlink in that directory to a real directory), and there is an account called "ftp" (which does not need to have a valid home directory). See **Virtual Servers** below.

Ftpd does a **chroot**(2) to the relevant base directory when an anonymous user logs in.

Note that **ftpd** allows remote users to log in as root if the password is known and -u not used.

Unusual Features

If a user's home directory is **/path/to/home/./**, FTP sessions under that UID will be chroot()ed. In addition, if a user's home directory is **/path/to/home/./directory** the session will be chroot()ed to /path/to/home and the FTP session will start in 'directory'.

As noted above, this **pure-ftpd** omits several features that are required by the RFC or might be considered useful at first. Here is a list of the most important omissions.

On-the-fly tar is not supported, for several reasons. I feel that users who want to get many files should use a special FTP client such as "mirror," which also supports incremental fetch. I don't want to either add several hundred lines of code to create tar files or execute an external tar. Finally, on-the-fly tar distorts log files.

On-the-fly compression is left out too. Most files on an FTP site are compressed already, and if a file isn't, there presumably is a reason why. (As for decompression: Don't FTP users waste bandwidth enough without help from on-the-fly decompression?)

Directory Aliases

Shortcuts for the "cd" command can be set up if the server has been compiled with the **--with-diraliases** feature.

To enable directory aliases, create a file called */etc/pure-ftpd/pureftpd-dir-aliases* and alternate lines of alias names and associated directories.

Anonymous Ftp

This server leaves out some of the commands and features that have been used to subvert anonymous FTP servers in the past, but still you have to be a little bit careful in order to support anonymous FTP without risk to the rest of your files.

Make *~ftp* and all files and directories below this directory owned by some user other than "ftp," and only the *.../incoming* directory/directories writable by "ftp." It is probably best if all directories are writable only by a special group such as "ftpadmin" and "ftp" is not a member of this group.

If you do not trust the local users, put *~ftp* on a separate partition, so local users can't hard-link unapproved files into the anonymous FTP area.

Use of the **-s** option is strongly suggested. (Simply add "-s" to the end of the **ftpd** line in */etc/inetd.conf* to enable it.)

Most other FTP servers require that a number of files such as *~ftp/bin/l*s exist. This server does not require that any files or directories within *~ftp* whatsoever exist, and I recommend that all such unnecessary files are removed (for no real reason).

It may be worth considering to run the anonymous FTP service as a virtual server, to get

automatic logins and to firewall off the FTP address/port to which real users can log in.

If your server is a public FTP site, you may want to allow only 'ftp' and 'anonymous' users to log in. Use the **-e** option for this. Real accounts will be ignored and you will get a secure, anonymous-only FTP server.

Magic Files

The files `<ftproot>/.banner` and `.message` are magical.

If there is a file called `.banner` in the root directory of the anonymous FTP area, or in the root directory of a virtual host, and it is shorter than 1024 bytes, it is printed upon login. (If the client does not log in explicitly, and an implicit login is triggered by a CWD or CDUP command, the banner is not printed. This is regrettable but hard to avoid.)

If there is a file called `.message` in any directory and it is shorter than 1024 bytes, that file is printed whenever a user enters that directory using CWD or CDUP.

Virtual Servers

You can run several different anonymous FTP servers on one host, by giving the host several IP addresses with different DNS names.

Here are the steps needed to create an extra server using an IP alias on linux 2.4.x, called "ftp.example.com" on address 10.11.12.13. on the IP alias eth0.

1. Create an "ftp" account if you do not have one. It is best if the account does not have a valid home directory and shell. I prefer to make `/dev/null` the ftp account's home directory and shell. **Ftpd** uses this account to set the anonymous users' uid.
2. Create a directory as described in **Anonymous FTP** and make a symlink called `/etc/pure-ftpd/pure-ftpd/10.11.12.13` which points to this directory.
3. Make sure your kernel has support for IP aliases.
4. Make sure that the following commands are run at boot:

```
/sbin/ifconfig eth0:1 10.11.12.13
```

That should be all. If you have problems, here are some things to try.

First, symlink `/etc/pure-ftpd/pure-ftpd/127.0.0.1` to some directory and say "ftp localhost". If that doesn't log you in, the problem is with **ftpd**.

If not, "ping -v 10.11.12.13" and/or "ping -v ftp.example.com" from the same host. If this does not work, the problem is with the IP alias.

Next, try "ping -v 10.11.12.13" from a host on the local ethernet, and afterwards

"/sbin/arp -a". If 10.11.12.13 is listed among the ARP entries with the correct hardware address, the problem is probably with the IP alias. If 10.11.12.13 is listed, but has hardware address 0:0:0:0:0:0, then proxy-ARP isn't working.

If none of that helps, I'm stumped. Good luck.

Warning: If you setup a virtual hosts, normal users will not be able to login via this name, so **don't** create link/directory in */etc/pure-ftpd/pure-ftpd* for your regular hostname.

Files

/etc/passwd is used via libc (and PAM in this case), to get the uid and home directory of normal users, the uid and home directory of "ftp" for normal anonymous ftp, and just the uid of "ftp" for virtual ftp hosts.

/etc/shadow is used like */etc/passwd* if shadow support is enabled.

/etc/group is used via libc, to get the group membership of normal users.

/proc/net/tcp is used to count existing FTP connections, if the **-c** or **-p** options are used

/etc/pure-ftpd/pure-ftpd/<ip address> is the base directory for the <ip address> virtual ftp server, or a symbolic link to its base directory. **Ftpd** does a **chroot**(2) into this directory when a user logs in to <ip address>, thus symlinks outside this directory will not work.

~ftp is the base directory for "normal" anonymous FTP. **Ftpd** does a **chroot**(2) into this directory when an anonymous user logs in, thus symlinks outside this directory will not work.

Ls

The behaviour of LIST and NLST is a tricky issue. Few servers send RFC-compliant responses to LIST, and some clients depend on non-compliant responses.

This server uses **glob**(3) to do filename globbing.

The response to NLST is by default similar to that of **ls**(1), and that to LIST is by default similar to that of **ls -l** or **ls -lg** on most Unix systems, except that the "total" count is meaningless. Only regular files, directories and symlinks are shown. Only important **ls** options are supported:

-1

Undoes **-l** and **-C**.

-a

lists even files/directories whose names begin with ".".

-C

lists files in as many columns as will fit on the screen. Undoes **-1** and **-l**.

-d

lists argument directories' names rather their contents.

-D

List files beginning with a dot ('.') even when the client doesn't append the **-a** option to the **list** command.

-F

appends '*' to executable regular files, '@' to symlinks and '/' to directories.

-l

shows various details about the file, including file group. See [ls\(1\)](#) for details. Undoes **-1** and **-C**.

-r

reverses the sorting order (modifies **-S** and **-t** and the default alphabetical ordering).

-R

recursively descends into subdirectories of the argument directories.

-S

Sorts by file size instead of by name. Undoes **-t**.

-t

Sorts by file modification time instead of by name. Undoes **-S**.

Protocol

Here are the FTP commands supported by this server.

**ABOR NOOP ALLO USER PASS QUIT SYST PORT EPRT PASV EPSV SPSV PWD
XPWD CWD XCWD CDUP XCUP HELP RETR REST DELE STOR APPE STOU MKD
XMKD RMD XRMD LIST NLST TYPE MODE STRU XDBG MDTM SIZE RNFR RNTD**

**STAT MLST MLSD FEAT ESTA ESTP AUTH TLS PBSZ PROT OPTS UTF8 OPTS MLST
SITE IDLE SITE CHMOD SITE HELP SITE TIME SITE UTIME**

Bugs

Please report bugs to the mailing-list (see below). Pure-FTPd looks very stable and is used on production servers. However it comes with no warranty and it can have nasty bugs or security flaws.

Home Page

<http://www.pureftpd.org/>

New Versions

See the mailing-list on <http://www.pureftpd.org/ml/>.

Author And License

Troll-FTPd was written by Arnt Gulbrandsen <agulbra@troll.no> and copyright 1995-2002 Troll Tech AS, Waldemar Thranes gate 98B, N-0175 Oslo, Norway, fax +47 22806380.

Pure-FTPd is (C)opyleft 2001-2010 by Frank DENIS <j at pureftpd dot org> and the Pure-FTPd team.

This software is covered by the BSD license.

Contributors:

Arnt Gulbrandsen, Troll Tech AS, Janos Farkas, August Fullford, Ximenes Zalteca, Patrick Michael Kane, Arkadiusz Miskiewicz, Michael K. Johnson, Kelley Lingerfelt, Sebastian Andersson, Andreas Westin, Jason Lunz, Mathias Gumz, Claudiu Costin, Ping, Paul Lasarev, Jean-Mathieux Schaffhauser, Emmanuel Hocdet, Sami Koskinen, Sami Farin, Luis Llorente Campo, Peter Pentchev, Darren Casey, The Regents of the University of California, Theo de Raadt (OpenBSD), Matthias Andree, Isak Lyberth, Steve Reid, RSA Data Security Inc, Trilucid, Dmitry Lebkov, Johan Huisman, Thorsten Kukuk, Jan van Veen, Roger Constantin Demetrescu, Stefano F., Robert Varga, Freeman, James Metcalf, Im Eunjea, Philip Gladstone, Kenneth Stailey, Brad Smith, Ulrik Sartipy, Cindy Marasco, Nicolas Doye, Thomas Briggs, Stanton Gallegos, Florin Andrei, Chan Wilson, Bjoern Metzendorf, Ben Gertzfield, Akhilesh Mritunjai, Dawid Szymanski, Kurt Inge Smadal, Alex Dupre, Gabriele Vinci, Andrey Ulanov, Fygyul Hether, Jeffrey Lim, Ying-Chieh Liao, Johannes Erdfelt, Martin Sarfy, Clive Goodhead, Aristoteles Pagaltzis, Stefan Hornburg, Mehmet Cokcevik, Brynjar Eide, Torgnt Wernersson, Banhalmi Csaba, Volodin D, Oriol MagranÃ©, Jui-Nan Lin, Patrick Gosling, Marc Balmer, Rajat Upadhyaya / Novell, Christian Cier-Zniewski, Wilco Baan Hofman.

See Also

[ftp\(1\)](#), **[pure-ftpd\(8\)](#)** **[pure-ftpwho\(8\)](#)** **[pure-mrtginfo\(8\)](#)** **[pure-uploadscript\(8\)](#)**
[pure-statsdecode\(8\)](#) **[pure-pw\(8\)](#)** **[pure-quotacheck\(8\)](#)** **[pure-authd\(8\)](#)**

RFC 959, **RFC 2228**, **RFC 2389** and **RFC 2428**.

Referenced By

[ccze\(1\)](#), **[pure-pwconvert\(8\)](#)**