NAME

ftpd **-** Internet File Transfer Protocol server

SYNOPSIS

ftpd [**-468ABDdEhMmOoRrSUvW**] [**-l** [**-l**]] [**-a** *address*] [**-P** *port*] [**-p** *file*] [**-T** *maxtimeout*] [**-t** *timeout*] [**-u** *umask*]

DEPRECATION NOTICE

The FreeBSD base system **ftpd** is deprecated, and will be removed in FreeBSD 15.0. Users are advised to install the *ftp/freebsd-ftpd* port or package instead.

DESCRIPTION

The **ftpd** utility is the Internet File Transfer Protocol server process. The server uses the TCP protocol and listens at the port specified with the **-P** option or in the "ftp" service specification; see services(5).

Available options:

-4 When **-D** is specified, accept connections via AF_INET socket.

-6 When **-D** is specified, accept connections via AF_INET6 socket.

-8 Enable transparent UTF-8 mode. RFC 2640 compliant clients will be told that the character encoding used by the server is UTF-8, which is the only effect of the option.

This option does not enable any encoding conversion for server file names; it implies instead that the names of files on the server are encoded in UTF-8. As for files uploaded via FTP, it is the duty of the RFC 2640 compliant client to convert their names from the client's local encoding to UTF-8. FTP command names and own **ftpd** messages are always encoded in ASCII, which is a subset of UTF-8. Hence no need for server-side conversion at all.

-A Allow only anonymous ftp access.

-a When **-D** is specified, accept connections only on the specified *address*.

-B With this option set, **ftpd** sends authentication success and failure messages to the blacklistd(8) daemon. If this option is not specified, no communcation with the blacklistd(8) daemon is attempted.

-D With this option set, **ftpd** will detach and become a daemon, accepting connections on the FTP port and forking children processes to handle them. This is lower overhead than starting **ftpd**

from inetd(8) and is thus useful on busy servers to reduce load.

-d Debugging information is written to the syslog using LOG_FTP.

-E Disable the EPSV command. This is useful for servers behind older firewalls.

-h Disable printing host-specific information, such as the server software version or hostname, in server messages.

-l Each successful and failed ftp(1) session is logged using syslog with a facility of LOG_FTP. If this option is specified twice, the retrieve (get), store (put), append, delete, make directory, remove directory and rename operations and their filename arguments are also logged. By default, syslogd(8) logs these to */var/log/xferlog*.

-M Prevent anonymous users from creating directories.

-m Permit anonymous users to overwrite or modify existing files if allowed by file system permissions. By default, anonymous users cannot modify existing files; in particular, files to upload will be created under a unique name.

-O Put server in write-only mode for anonymous users only. RETR is disabled for anonymous users, preventing anonymous downloads. This has no effect if **-o** is also specified.

-o Put server in write-only mode. RETR is disabled, preventing downloads.

-P When **-D** is specified, accept connections at *port*, specified as a numeric value or service name, instead of at the default "ftp" port.

-p When **-D** is specified, write the daemon’s process ID to *file* instead of the default pid file, */var/run/ftpd.pid*.

-R With this option set, **ftpd** will revert to historical behavior with regard to security checks on user operations and restrictions on PORT requests. Currently, **ftpd** will only honor PORT commands directed to unprivileged ports on the remote user’s host (which violates the FTP protocol specification but closes some security holes).

-r Put server in read-only mode. All commands which may modify the local file system are disabled.

-S With this option set, **ftpd** logs all anonymous file downloads to the file */var/log/ftpd* when this file exists.

-T A client may also request a different timeout period; the maximum period allowed may be set to *timeout* seconds with the **-T** option. The default limit is 2 hours.

-t The inactivity timeout period is set to *timeout* seconds (the default is 15 minutes).

-U This option instructs ftpd to use data ports in the range of IP_PORTRANGE_DEFAULT instead of in the range of IP_PORTRANGE_HIGH. Such a change may be useful for some specific firewall configurations; see ip(4) for more information.

Note that option is a virtual no-op in FreeBSD 5.0 and above; both port ranges are identical by default.

-u The default file creation mode mask is set to *umask*, which is expected to be an octal numeric value. Refer to umask(2) for details. This option may be overridden by login.conf(5).

-v A synonym for **-d**.

-W Do not log FTP sessions to the user accounting database.

The file */var/run/nologin* can be used to disable ftp access. If the file exists, **ftpd** displays it and exits. If the file */etc/ftpwelcome* exists, **ftpd** prints it before issuing the "ready" message. If the file */etc/ftpmotd* exists, **ftpd** prints it after a successful login. Note the motd file used is the one relative to the login environment. This means the one in *~ftp/etc* in the anonymous user’s case.

The ftp server currently supports the following ftp requests. The case of the requests is ignored. Requests marked [RW] are disabled if **-r** is specified.

Request	**Description**
ABOR	abort previous command
ACCT	specify account (ignored)
ALLO	allocate storage (vacuously)
APPE	append to a file [RW]
CDUP	change to parent of current working directory
CWD	change working directory
DELE	delete a file [RW]
EPRT	specify data connection port, multiprotocol
EPSV	prepare for server-to-server transfer, multiprotocol
FEAT	give information on extended features of server

HELP	give help information
LIST	give list files in a directory ("ls -lgA")
LPRT	specify data connection port, multiprotocol
LPSV	prepare for server-to-server transfer, multiprotocol
MDTM	show last modification time of file
MKD	make a directory [RW]
MODE	specify data transfer *mode*
NLST	give name list of files in directory
NOOP	do nothing
PASS	specify password
PASV	prepare for server-to-server transfer
PORT	specify data connection port
PWD	print the current working directory
QUIT	terminate session
REST	restart incomplete transfer
RETR	retrieve a file
RMD	remove a directory [RW]
RNFR	specify rename-from file name [RW]
RNTO	specify rename-to file name [RW]
SITE	non-standard commands (see next section)
SIZE	return size of file
STAT	return status of server
STOR	store a file [RW]
STOU	store a file with a unique name [RW]
STRU	specify data transfer *structure*
SYST	show operating system type of server system
TYPE	specify data transfer *type*
USER	specify user name
XCUP	change to parent of current working directory (deprecated)
XCWD	change working directory (deprecated)
XMKD	make a directory (deprecated) [RW]
XPWD	print the current working directory (deprecated)
XRMD	remove a directory (deprecated) [RW]

The following non-standard or UNIX specific commands are supported by the SITE request.

Request	**Description**
UMASK	change umask, e.g. ‘‘SITE UMASK 002’’
IDLE	set idle-timer, e.g. ‘‘SITE IDLE 60’’
CHMOD	change mode of a file [RW], e.g. ‘‘SITE CHMOD 755 filename’’

MD5	report the files MD5 checksum, e.g. ‘‘SITE MD5 filename’’
HELP	give help information

Note: SITE requests are disabled in case of anonymous logins.

The remaining ftp requests specified in Internet RFC 959 are recognized, but not implemented. MDTM and SIZE are not specified in RFC 959, but will appear in the next updated FTP RFC. To avoid possible denial-of-service attacks, SIZE requests against files larger than 10240 bytes will be denied if the current transfer type is ASCII.

The ftp server will abort an active file transfer only when the ABOR command is preceded by a Telnet "Interrupt Process" (IP) signal and a Telnet "Synch" signal in the command Telnet stream, as described in Internet RFC 959. If a STAT command is received during a data transfer, preceded by a Telnet IP and Synch, transfer status will be returned.

The **ftpd** utility interprets file names according to the "globbing" conventions used by csh(1). This allows users to utilize the metacharacters "*?[]{}~".

The **ftpd** utility authenticates users according to six rules.

1. The login name must be in the password data base and not have a null password. In this case a password must be provided by the client before any file operations may be performed.

2. The login name must not appear in the file */etc/ftpusers*.

3. The login name must not be a member of a group specified in the file */etc/ftpusers*. Entries in this file interpreted as group names are prefixed by an "at" ‘@’ sign.

4. The user must have a standard shell returned by getusershell(3).

5. If the user name appears in the file */etc/ftpchroot*, or the user is a member of a group with a group entry in this file, i.e., one prefixed with ‘@’, the session’s root will be changed to the directory specified in this file or to the user’s login directory by chroot(2) as for an "anonymous" or "ftp" account (see next item). See ftpchroot(5) for a detailed description of the format of this file. This facility may also be triggered by enabling the boolean "ftp-chroot" capability in login.conf(5). However, the user must still supply a password. This feature is intended as a compromise between a fully anonymous account and a fully privileged account. The account should also be set up as for an anonymous account.

6. If the user name is "anonymous" or "ftp", an anonymous ftp account must be present in the

password file (user "ftp"). In this case the user is allowed to log in by specifying any password (by convention an email address for the user should be used as the password). When the **-S** option is set, all transfers are logged as well.

In the last case, **ftpd** takes special measures to restrict the client’s access privileges. The server performs a chroot(2) to the home directory of the "ftp" user. As a special case if the "ftp" user’s home directory pathname contains the */./* separator, **ftpd** uses its left-hand side as the name of the directory to do chroot(2) to, and its right-hand side to change the current directory to afterwards. A typical example for this case would be */var/spool/ftp/./pub*. In order that system security is not breached, it is recommended that the "ftp" subtree be constructed with care, following these rules:

~ftp Make the home directory owned by "root" and unwritable by anyone.

~ftp/etc Make this directory owned by "root" and unwritable by anyone (mode 555). The files pwd.db (see passwd(5)) and group(5) must be present for the ls(1) command to be able to produce owner names rather than numbers. The password field in passwd(5) is not used, and should not contain real passwords. The file *ftpmotd*, if present, will be printed after a successful login. These files should be mode 444.

~ftp/pub This directory and the subdirectories beneath it should be owned by the users and groups responsible for placing files in them, and be writable only by them (mode 755 or 775). They should *not* be owned or writable by "ftp" or its group, otherwise guest users can fill the drive with unwanted files.

If the system has multiple IP addresses, **ftpd** supports the idea of virtual hosts, which provides the ability to define multiple anonymous ftp areas, each one allocated to a different internet address. The file */etc/ftphosts* contains information pertaining to each of the virtual hosts. Each host is defined on its own line which contains a number of fields separated by whitespace:

hostname Contains the hostname or IP address of the virtual host.

user Contains a user record in the system password file. As with normal anonymous ftp, this user’s access uid, gid and group memberships determine file access to the anonymous ftp area. The anonymous ftp area (to which any user is chrooted on login) is determined by the home directory defined for the account. User id and group for any ftp account may be the same as for the standard ftp user.

statfile File to which all file transfers are logged, which defaults to */var/log/ftpd*.

welcome This file is the welcome message displayed before the server ready prompt. It defaults to */etc/ftpwelcome*.

motd	This file is displayed after the user logs in. It defaults to */etc/ftpmotd*.

Lines beginning with a ’#’ are ignored and can be used to include comments.

Defining a virtual host for the primary IP address or hostname changes the default for ftp logins to that address. The ’user’, ’statfile’, ’welcome’ and ’motd’ fields may be left blank, or a single hyphen ’-’ used to indicate that the default value is to be used.

As with any anonymous login configuration, due care must be given to setup and maintenance to guard against security related problems.

The **ftpd** utility has internal support for handling remote requests to list files, and will not execute */bin/ls* in either a chrooted or non-chrooted environment. The *~/bin/ls* executable need not be placed into the chrooted tree, nor need the *~/bin* directory exist.

FILES

/etc/ftpusers	List of unwelcome/restricted users.
/etc/ftpchroot	List of normal users who should be chroot’d.
/etc/ftphosts	Virtual hosting configuration file.
/etc/ftpwelcome	Welcome notice.
/etc/ftpmotd	Welcome notice after login.
/var/run/ftpd.pid	Default pid file for daemon mode.
/var/run/nologin	Displayed and access refused.
/var/log/ftpd	Log file for anonymous transfers.
/var/log/xferlog	Default place for session logs.
/var/spool/ftp	Recommended directory for the FTP root directory (the home directory of the ftp user).

SEE ALSO

ftp(1), umask(2), getusershell(3), ftpchroot(5), login.conf(5), inetd(8), syslogd(8)

HISTORY

The **ftpd** utility appeared in 4.2BSD. IPv6 support was added in WIDE Hydrangea IPv6 stack kit.

BUGS

The server must run as the super-user to create sockets with privileged port numbers. It maintains an effective user id of the logged in user, reverting to the super-user only when binding addresses to sockets. The possible security holes have been extensively scrutinized, but are possibly incomplete.