**File Transfer Protocol**

From Wikipedia, the free encyclopedia

[Jump to navigation](https://en.wikipedia.org/wiki/File_Transfer_Protocol#mw-head) [Jump to search](https://en.wikipedia.org/wiki/File_Transfer_Protocol#searchInput)

"FTP" redirects here. For other uses, see [FTP (disambiguation)](https://en.wikipedia.org/wiki/FTP_(disambiguation)).

|  |
| --- |
| [**Internet protocol suite**](https://en.wikipedia.org/wiki/Internet_protocol_suite) |
| [**Application layer**](https://en.wikipedia.org/wiki/Application_layer) |
| * [BGP](https://en.wikipedia.org/wiki/Border_Gateway_Protocol) * [DHCP](https://en.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol) * [DNS](https://en.wikipedia.org/wiki/Domain_Name_System) * FTP * [HTTP](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) * [HTTPS](https://en.wikipedia.org/wiki/HTTPS) * [IMAP](https://en.wikipedia.org/wiki/Internet_Message_Access_Protocol) * [LDAP](https://en.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol) * [MGCP](https://en.wikipedia.org/wiki/Media_Gateway_Control_Protocol) * [MQTT](https://en.wikipedia.org/wiki/MQTT) * [NNTP](https://en.wikipedia.org/wiki/Network_News_Transfer_Protocol) * [NTP](https://en.wikipedia.org/wiki/Network_Time_Protocol) * [POP](https://en.wikipedia.org/wiki/Post_Office_Protocol) * [PTP](https://en.wikipedia.org/wiki/Precision_Time_Protocol) * [ONC/RPC](https://en.wikipedia.org/wiki/Open_Network_Computing_Remote_Procedure_Call) * [RTP](https://en.wikipedia.org/wiki/Real-time_Transport_Protocol) * [RTSP](https://en.wikipedia.org/wiki/Real_Time_Streaming_Protocol) * [RIP](https://en.wikipedia.org/wiki/Routing_Information_Protocol) * [SIP](https://en.wikipedia.org/wiki/Session_Initiation_Protocol) * [SMTP](https://en.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol) * [SNMP](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol) * [SSH](https://en.wikipedia.org/wiki/Secure_Shell) * [Telnet](https://en.wikipedia.org/wiki/Telnet) * [TLS/SSL](https://en.wikipedia.org/wiki/Transport_Layer_Security) * [XMPP](https://en.wikipedia.org/wiki/XMPP) * [*more...*](https://en.wikipedia.org/wiki/Category:Application_layer_protocols) |
| [**Transport layer**](https://en.wikipedia.org/wiki/Transport_layer) |
| * [TCP](https://en.wikipedia.org/wiki/Transmission_Control_Protocol) * [UDP](https://en.wikipedia.org/wiki/User_Datagram_Protocol) * [DCCP](https://en.wikipedia.org/wiki/Datagram_Congestion_Control_Protocol) * [SCTP](https://en.wikipedia.org/wiki/Stream_Control_Transmission_Protocol) * [RSVP](https://en.wikipedia.org/wiki/Resource_Reservation_Protocol) * [*more...*](https://en.wikipedia.org/wiki/Category:Transport_layer_protocols) |
| [**Internet layer**](https://en.wikipedia.org/wiki/Internet_layer) |
| * [IP](https://en.wikipedia.org/wiki/Internet_Protocol)   + [IPv4](https://en.wikipedia.org/wiki/IPv4)   + [IPv6](https://en.wikipedia.org/wiki/IPv6) * [ICMP](https://en.wikipedia.org/wiki/Internet_Control_Message_Protocol) * [ICMPv6](https://en.wikipedia.org/wiki/Internet_Control_Message_Protocol_for_IPv6) * [ECN](https://en.wikipedia.org/wiki/Explicit_Congestion_Notification) * [IGMP](https://en.wikipedia.org/wiki/Internet_Group_Management_Protocol) * [IPsec](https://en.wikipedia.org/wiki/IPsec) * [*more...*](https://en.wikipedia.org/wiki/Category:Internet_layer_protocols) |
| [**Link layer**](https://en.wikipedia.org/wiki/Link_layer) |
| * [ARP](https://en.wikipedia.org/wiki/Address_Resolution_Protocol) * [NDP](https://en.wikipedia.org/wiki/Neighbor_Discovery_Protocol) * [OSPF](https://en.wikipedia.org/wiki/Open_Shortest_Path_First) * [Tunnels](https://en.wikipedia.org/wiki/Tunneling_protocol)   + [L2TP](https://en.wikipedia.org/wiki/Layer_2_Tunneling_Protocol) * [PPP](https://en.wikipedia.org/wiki/Point-to-Point_Protocol) * [MAC](https://en.wikipedia.org/wiki/Medium_access_control)   + [Ethernet](https://en.wikipedia.org/wiki/Ethernet)   + [Wi-Fi](https://en.wikipedia.org/wiki/Wi-Fi)   + [DSL](https://en.wikipedia.org/wiki/Digital_subscriber_line)   + [ISDN](https://en.wikipedia.org/wiki/Integrated_Services_Digital_Network)   + [FDDI](https://en.wikipedia.org/wiki/Fiber_Distributed_Data_Interface) * [*more...*](https://en.wikipedia.org/wiki/Category:Link_protocols) |
| * [v](https://en.wikipedia.org/wiki/Template:IPstack) * [t](https://en.wikipedia.org/wiki/Template_talk:IPstack) * [e](https://en.wikipedia.org/w/index.php?title=Template:IPstack&action=edit) |

The **File Transfer Protocol** (**FTP**) is a standard [network protocol](https://en.wikipedia.org/wiki/Network_protocol) used for the transfer of [computer files](https://en.wikipedia.org/wiki/Computer_file) between [a client and server](https://en.wikipedia.org/wiki/Client%E2%80%93server_model) on a [computer network](https://en.wikipedia.org/wiki/Computer_network).

FTP is built on a client-server model architecture using separate control and data connections between the client and the server.[[1]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-for-1) FTP users may authenticate themselves with a [clear-text](https://en.wikipedia.org/wiki/Clear_text) sign-in protocol, normally in the form of a username and password, but can connect anonymously if the server is configured to allow it. For secure transmission that protects the username and password, and encrypts the content, FTP is often [secured](https://en.wikipedia.org/wiki/File_Transfer_Protocol#Security) with [SSL/TLS](https://en.wikipedia.org/wiki/Transport_Layer_Security) ([FTPS](https://en.wikipedia.org/wiki/FTPS)) or replaced with [SSH File Transfer Protocol](https://en.wikipedia.org/wiki/SSH_File_Transfer_Protocol) (SFTP).

The first FTP client applications were [command-line programs](https://en.wikipedia.org/wiki/Command-line_interface) developed before [operating systems](https://en.wikipedia.org/wiki/Operating_system) had [graphical user interfaces](https://en.wikipedia.org/wiki/Graphical_user_interface), and are still shipped with most [Windows](https://en.wikipedia.org/wiki/Windows), [Unix](https://en.wikipedia.org/wiki/Unix), and [Linux](https://en.wikipedia.org/wiki/Linux) operating systems.[[2]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-tcpip-2)[[3]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-net+-3) Many FTP clients and automation utilities have since been developed for [desktops](https://en.wikipedia.org/wiki/Desktop_computer), servers, mobile devices, and hardware, and FTP has been incorporated into productivity applications, such as [HTML editors](https://en.wikipedia.org/wiki/HTML_editor).



**Contents**

* [1 History of FTP servers](https://en.wikipedia.org/wiki/File_Transfer_Protocol#History_of_FTP_servers)
* [2 Protocol overview](https://en.wikipedia.org/wiki/File_Transfer_Protocol#Protocol_overview)
  + [2.1 Communication and data transfer](https://en.wikipedia.org/wiki/File_Transfer_Protocol#Communication_and_data_transfer)
  + [2.2 NAT and firewall traversal](https://en.wikipedia.org/wiki/File_Transfer_Protocol#NAT_and_firewall_traversal)
  + [2.3 Data types](https://en.wikipedia.org/wiki/File_Transfer_Protocol#Data_types)
  + [2.4 File structures](https://en.wikipedia.org/wiki/File_Transfer_Protocol#File_structures)
  + [2.5 Data transfer modes](https://en.wikipedia.org/wiki/File_Transfer_Protocol#Data_transfer_modes)
* [3 Login](https://en.wikipedia.org/wiki/File_Transfer_Protocol#Login)
  + [3.1 Anonymous FTP](https://en.wikipedia.org/wiki/File_Transfer_Protocol#Anonymous_FTP)
* [4 Differences from HTTP](https://en.wikipedia.org/wiki/File_Transfer_Protocol#Differences_from_HTTP)
* [5 Web browser support](https://en.wikipedia.org/wiki/File_Transfer_Protocol#Web_browser_support)
  + [5.1 Syntax](https://en.wikipedia.org/wiki/File_Transfer_Protocol#Syntax)
* [6 Security](https://en.wikipedia.org/wiki/File_Transfer_Protocol#Security)
  + [6.1 FTP over SSH](https://en.wikipedia.org/wiki/File_Transfer_Protocol#FTP_over_SSH)
* [7 Derivatives](https://en.wikipedia.org/wiki/File_Transfer_Protocol#Derivatives)
  + [7.1 FTPS](https://en.wikipedia.org/wiki/File_Transfer_Protocol#FTPS)
  + [7.2 SSH File Transfer Protocol](https://en.wikipedia.org/wiki/File_Transfer_Protocol#SSH_File_Transfer_Protocol)
  + [7.3 Trivial File Transfer Protocol](https://en.wikipedia.org/wiki/File_Transfer_Protocol#Trivial_File_Transfer_Protocol)
  + [7.4 Simple File Transfer Protocol](https://en.wikipedia.org/wiki/File_Transfer_Protocol#Simple_File_Transfer_Protocol)
* [8 FTP commands](https://en.wikipedia.org/wiki/File_Transfer_Protocol#FTP_commands)
* [9 FTP reply codes](https://en.wikipedia.org/wiki/File_Transfer_Protocol#FTP_reply_codes)
* [10 See also](https://en.wikipedia.org/wiki/File_Transfer_Protocol#See_also)
* [11 References](https://en.wikipedia.org/wiki/File_Transfer_Protocol#References)
* [12 Further reading](https://en.wikipedia.org/wiki/File_Transfer_Protocol#Further_reading)
* [13 External links](https://en.wikipedia.org/wiki/File_Transfer_Protocol#External_links)

**History of FTP servers**

The original specification for the File Transfer Protocol was written by [Abhay Bhushan](https://en.wikipedia.org/wiki/Abhay_Bhushan) and published as [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [114](https://tools.ietf.org/html/rfc114) on 16 April 1971. Until 1980, FTP ran on [NCP](https://en.wikipedia.org/wiki/Network_Control_Program), the predecessor of [TCP/IP](https://en.wikipedia.org/wiki/Internet_protocol_suite).[[2]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-tcpip-2) The protocol was later replaced by a TCP/IP version, [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [765](https://tools.ietf.org/html/rfc765) (June 1980) and [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [959](https://tools.ietf.org/html/rfc959) (October 1985), the current specification. Several proposed standards amend [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [959](https://tools.ietf.org/html/rfc959), for example [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [1579](https://tools.ietf.org/html/rfc1579) (February 1994) enables Firewall-Friendly FTP (passive mode), [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [2228](https://tools.ietf.org/html/rfc2228) (June 1997) proposes security extensions, [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [2428](https://tools.ietf.org/html/rfc2428) (September 1998) adds support for [IPv6](https://en.wikipedia.org/wiki/IPv6) and defines a new type of passive mode.[[4]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-clark-4)

**Protocol overview**

**Communication and data transfer**

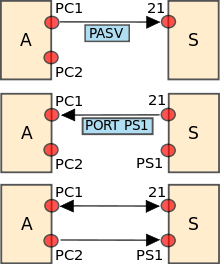
[](https://en.wikipedia.org/wiki/File:Passive_FTP_Verbindung.svg)

Illustration of starting a passive connection using port 21

FTP may run in *active* or *passive* mode, which determines how the data connection is established.[[5]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-slack-5) (Note, somewhat confusingly, this sense of "mode" is *different* from that of the MODE command in the FTP protocol, and actually corresponds to the PORT/PASV/EPSV/etc commands instead.) In both cases, the client creates a TCP control connection from a [random, usually an unprivileged](https://en.wikipedia.org/wiki/Ephemeral_port), [port](https://en.wikipedia.org/wiki/Port_(computer_networking)) N to the FTP server command port 21.

* In active mode, the client starts listening for incoming data connections from the server on port M. It sends the FTP command PORT M to inform the server on which port it is listening. The server then initiates a data channel to the client from its port 20, the FTP server data port.
* In situations where the client is behind a [firewall](https://en.wikipedia.org/wiki/Firewall_(computing)) and unable to accept incoming TCP connections, *passive mode* may be used. In this mode, the client uses the control connection to send a PASV command to the server and then receives a server IP address and server port number from the server,[[5]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-slack-5) which the client then uses to open a data connection from an arbitrary client port to the server IP address and server port number received.[[6]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-postel-6)

Both modes were updated in September 1998 to support [IPv6](https://en.wikipedia.org/wiki/IPv6). Further changes were introduced to the passive mode at that time, updating it to *extended passive mode*.[[7]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-7)

The server responds over the control connection with [three-digit status codes](https://en.wikipedia.org/wiki/List_of_FTP_server_return_codes) in ASCII with an optional text message. For example, "200" (or "200 OK") means that the last command was successful. The numbers represent the code for the response and the optional text represents a human-readable explanation or request (e.g. <Need account for storing file>).[[1]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-for-1) An ongoing transfer of file data over the data connection can be aborted using an interrupt message sent over the control connection.

The reason why FTP needs two ports (one for sending and one for receiving) is due to the fact that it was originally designed to operate on [Network Control Program](https://en.wikipedia.org/wiki/Network_Control_Program) (NCP), which was a [simplex protocol](https://en.wikipedia.org/wiki/Simplex_communication) that utilized two [port addresses](https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers), establishing two connections, for two-way communications. An odd and an even port were reserved for each [application layer](https://en.wikipedia.org/wiki/Application_layer) application or protocol. The standardization of TCP and UDP reduced the need for the use of two simplex ports for each application down to one duplex port,[[8]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-stevens1994v1-8):15 but the FTP protocol was never altered to only use one port, but continued using two for backwards compatibility.

**NAT and firewall traversal**

FTP normally transfers data by having the server connect back to the client, after the PORT command is sent by the client. This is problematic for both [NATs](https://en.wikipedia.org/wiki/Network_address_translation) and firewalls, which do not allow connections from the Internet towards internal hosts.[[9]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-ncftp-9) For NATs, an additional complication is that the representation of the IP addresses and port number in the PORT command refer to the internal host's IP address and port, rather than the public IP address and port of the NAT.

There are two approaches to solve this problem. One is that the FTP client and FTP server use the PASV command, which causes the data connection to be established from the FTP client to the server.[[9]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-ncftp-9) This is widely used by modern FTP clients. Another approach is for the NAT to alter the values of the PORT command, using an [application-level gateway](https://en.wikipedia.org/wiki/Application-level_gateway) for this purpose.[[9]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-ncftp-9)

**Data types**

While transferring data over the network, four data types are defined:[[2]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-tcpip-2)[[3]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-net+-3)[[4]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-clark-4)

* [ASCII](https://en.wikipedia.org/wiki/ASCII) (TYPE A): Used for text. Data is converted, if needed, from the sending host's character representation to ["8-bit ASCII"](https://en.wikipedia.org/wiki/Extended_ASCII) before transmission, and (again, if necessary) to the receiving host's character representation. As a consequence, this mode is inappropriate for files that contain data other than plain text.
* Image (TYPE I, commonly called [Binary](https://en.wikipedia.org/wiki/Binary_data) mode): The sending machine sends each file [byte](https://en.wikipedia.org/wiki/Byte) by byte, and the recipient stores the [bytestream](https://en.wikipedia.org/wiki/Bytestream) as it receives it. (Image mode support has been recommended for all implementations of FTP).
* [EBCDIC](https://en.wikipedia.org/wiki/EBCDIC) (TYPE E): Used for plain text between hosts using the EBCDIC character set.
* Local (TYPE L *n*): Designed to support file transfer between machines which do not use 8-bit bytes, e.g. [36-bit systems](https://en.wikipedia.org/wiki/36-bit_computing) such as DEC [PDP-10s](https://en.wikipedia.org/wiki/PDP-10). For example, "TYPE L 9" would be used to transfer data in 9-bit bytes, or "TYPE L 36" to transfer 36-bit words. Most contemporary FTP clients/servers only support L 8, which is equivalent to I.

An expired [Internet Draft](https://en.wikipedia.org/wiki/Internet_Draft) defined a TYPE U for transferring [Unicode](https://en.wikipedia.org/wiki/Unicode) text files using [UTF-8](https://en.wikipedia.org/wiki/UTF-8);[[10]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-10) although the draft never became an RFC, it has been implemented by several FTP clients/servers.

Note these data types are commonly called "modes", although ambiguously that word is also used to refer to active-vs-passive communication mode (see above), and the modes set by the FTP protocol MODE command (see below).

For text files (TYPE A and TYPE E), three different format control options are provided, to control how the file would be printed:

* Non-print (TYPE A N and TYPE E N) – the file does not contain any carriage control characters intended for a printer
* [Telnet](https://en.wikipedia.org/wiki/Telnet) (TYPE A T and TYPE E T) – the file contains Telnet (or in other words, ASCII C0) carriage control characters (CR, LF, etc)
* [ASA](https://en.wikipedia.org/wiki/ASA_carriage_control_characters) (TYPE A A and TYPE E A) – the file contains ASA carriage control characters

These formats were mainly relevant to [line printers](https://en.wikipedia.org/wiki/Line_printer); most contemporary FTP clients/servers only support the default format control of N.

**File structures**

File organization is specified using the STRU command. The following file structures are defined in section 3.1.1 of RFC959:

* **F** or FILE structure (stream-oriented). Files are viewed as an arbitrary sequence of bytes, characters or words. This is the usual file structure on Unix systems and other systems such as CP/M, MSDOS and Microsoft Windows. (Section 3.1.1.1)
* **R** or RECORD structure (record-oriented). Files are viewed as divided into records, which may be fixed or variable length. This file organization is common on mainframe and midrange systems, such as MVS, VM/CMS, OS/400 and VMS, which support [record-oriented filesystems](https://en.wikipedia.org/wiki/Record-oriented_filesystem).
* **P** or PAGE structure (page-oriented). Files are divided into pages, which may either contain data or metadata; each page may also have a header giving various attributes. This file structure was specifically designed for [TENEX](https://en.wikipedia.org/wiki/TENEX_(operating_system)) systems, and is generally not supported on other platforms. RFC1123 section 4.1.2.3 recommends that this structure not be implemented.

Most contemporary FTP clients and servers only support STRU F. STRU R is still in use in mainframe and minicomputer file transfer applications.

**Data transfer modes**

Data transfer can be done in any of three modes:[[1]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-for-1)[[2]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-tcpip-2)

* Stream mode (MODE S): Data is sent as a continuous stream, relieving FTP from doing any processing. Rather, all processing is left up to [TCP](https://en.wikipedia.org/wiki/Transmission_Control_Protocol). No End-of-file indicator is needed, unless the data is divided into [records](https://en.wikipedia.org/wiki/Record_(computer_science)).
* Block mode (MODE B): Designed primarily for transferring record-oriented files (STRU R), although can also be used to transfer stream-oriented (STRU F) text files. FTP puts each record (or line) of data into several blocks (block header, byte count, and data field) and then passes it on to TCP.[[4]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-clark-4)
* Compressed mode (MODE C): Extends MODE B with data compression using [run-length encoding](https://en.wikipedia.org/wiki/Run-length_encoding).

Most contemporary FTP clients and servers do not implement MODE B or MODE C; FTP clients and servers for mainframe and minicomputer operating systems are the exception to that.

Some FTP software also implements a [DEFLATE](https://en.wikipedia.org/wiki/DEFLATE)-based compressed mode, sometimes called "Mode Z" after the command that enables it. This mode was described in an [Internet Draft](https://en.wikipedia.org/wiki/Internet_Draft), but not standardized.[[11]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-11)

[GridFTP](https://en.wikipedia.org/wiki/GridFTP) defines additional modes, MODE E[[12]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-12) and MODE X,[[13]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-13) as extensions of MODE B.

**Login**

FTP login uses normal username and password scheme for granting access.[[2]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-tcpip-2) The username is sent to the server using the USER command, and the password is sent using the PASS command.[[2]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-tcpip-2) This sequence is unencrypted "on the wire", so may be vulnerable to a network [sniffing attack](https://en.wikipedia.org/wiki/Sniffing_attack).[[14]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-14) If the information provided by the client is accepted by the server, the server will send a greeting to the client and the session will commence.[[2]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-tcpip-2) If the server supports it, users may log in without providing login credentials, but the same server may authorize only limited access for such sessions.[[2]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-tcpip-2)

**Anonymous FTP**

A host that provides an FTP service may provide [anonymous](https://en.wikipedia.org/wiki/Anonymity) FTP access.[[2]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-tcpip-2) Users typically log into the service with an 'anonymous' (lower-case and case-sensitive in some FTP servers) account when prompted for user name. Although users are commonly asked to send their [email](https://en.wikipedia.org/wiki/Email) address instead of a password,[[3]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-net+-3) no verification is actually performed on the supplied data.[[15]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-rfc16-15) Many FTP hosts whose purpose is to provide software updates will allow anonymous logins.[[3]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-net+-3)

**Differences from HTTP**

[HTTP](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) essentially fixes the bugs in FTP that made it inconvenient to use for many small ephemeral transfers as are typical in web pages.

FTP has a stateful control connection which maintains a current working directory and other flags, and each transfer requires a secondary connection through which the data are transferred. In "passive" mode this secondary connection is from client to server, whereas in the default "active" mode this connection is from server to client. This apparent role reversal when in active mode, and random port numbers for all transfers, is why firewalls and NAT gateways have such a hard time with FTP. HTTP is stateless and multiplexes control and data over a single connection from client to server on well-known port numbers, which trivially passes through NAT gateways and is simple for firewalls to manage.

Setting up an FTP control connection is quite slow due to the round-trip delays of sending all of the required commands and awaiting responses, so it is customary to bring up a control connection and hold it open for multiple file transfers rather than drop and re-establish the session afresh each time. In contrast, HTTP originally dropped the connection after each transfer because doing so was so cheap. While HTTP has subsequently gained the ability to reuse the TCP connection for multiple transfers, the conceptual model is still of independent requests rather than a session.

When FTP is transferring over the data connection, the control connection is idle. If the transfer takes too long, the firewall or NAT may decide that the control connection is dead and stop tracking it, effectively breaking the connection and confusing the download. The single HTTP connection is only idle between requests and it is normal and expected for such connections to be dropped after a time-out.

**Web browser support**

Most common [web browsers](https://en.wikipedia.org/wiki/Web_browser) can retrieve files hosted on FTP servers, although they may not support protocol extensions such as [FTPS](https://en.wikipedia.org/wiki/FTPS).[[3]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-net+-3)[[16]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-matthews-16) When an FTP—rather than an HTTP—[URL](https://en.wikipedia.org/wiki/URL) is supplied, the accessible contents on the remote server are presented in a manner that is similar to that used for other web content. A full-featured FTP client can be run within [Firefox](https://en.wikipedia.org/wiki/Firefox) in the form of an extension called [FireFTP](https://en.wikipedia.org/wiki/FireFTP).

As of 2019, major browsers such as Chrome and Firefox are deprecating FTP support to varying degrees,[[17]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-17) with Google planning to remove it entirely by Chrome 82. Mozilla is currently discussing proposals, including only removing support for old FTP implementations that are no longer in use to simplify their code.[[18]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-18)[[19]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-19)

**Syntax**

FTP URL syntax is described in [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [1738](https://tools.ietf.org/html/rfc1738), taking the form: ftp://[user[:password]@]host[:port]/url-path (the bracketed parts are optional).

For example, the URL ftp://public.ftp-servers.example.com/mydirectory/myfile.txt represents the file *myfile.txt* from the directory *mydirectory* on the server *public.ftp-servers.example.com* as an FTP resource. The URL ftp://user001:secretpassword@private.ftp-servers.example.com/mydirectory/myfile.txt adds a specification of the username and password that must be used to access this resource.

More details on specifying a username and password may be found in the browsers' documentation (e.g., [Firefox](https://en.wikipedia.org/wiki/Firefox)[[20]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-20) and [Internet Explorer](https://en.wikipedia.org/wiki/Internet_Explorer)[[21]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-21)). By default, most web browsers use passive (PASV) mode, which more easily traverses end-user firewalls.

Some variation has existed in how different browsers treat path resolution in cases where there is a non-root home directory for a user.[[22]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-22)

**Security**

FTP was not designed to be a secure protocol, and has many security weaknesses.[[23]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-nurdle-23) In May 1999, the authors of [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [2577](https://tools.ietf.org/html/rfc2577) listed a vulnerability to the following problems:

* [Brute-force attack](https://en.wikipedia.org/wiki/Brute-force_attack)
* [FTP bounce attack](https://en.wikipedia.org/wiki/FTP_bounce_attack)
* [Packet capture](https://en.wikipedia.org/wiki/Packet_capture)
* Port stealing (guessing the next open port and usurping a legitimate connection)
* [Spoofing attack](https://en.wikipedia.org/wiki/Spoofing_attack)
* Username enumeration
* [DoS or DDoS](https://en.wikipedia.org/wiki/Denial-of-service_attack)

FTP does not encrypt its traffic; all transmissions are in clear text, and usernames, passwords, commands and data can be read by anyone able to perform packet capture ([sniffing](https://en.wikipedia.org/wiki/Packet_analyzer)) on the network.[[2]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-tcpip-2)[[23]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-nurdle-23) This problem is common to many of the Internet Protocol specifications (such as [SMTP](https://en.wikipedia.org/wiki/SMTP), [Telnet](https://en.wikipedia.org/wiki/Telnet), POP and IMAP) that were designed prior to the creation of encryption mechanisms such as [TLS](https://en.wikipedia.org/wiki/Transport_Layer_Security) or SSL.[[4]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-clark-4)

Common solutions to this problem include:

1. Using the secure versions of the insecure protocols, e.g., [FTPS](https://en.wikipedia.org/wiki/FTPS) instead of FTP and TelnetS instead of Telnet.
2. Using a different, more secure protocol that can handle the job, e.g. [SSH File Transfer Protocol](https://en.wikipedia.org/wiki/SSH_File_Transfer_Protocol) or [Secure Copy Protocol](https://en.wikipedia.org/wiki/Secure_Copy_Protocol).
3. Using a secure tunnel such as [Secure Shell](https://en.wikipedia.org/wiki/Secure_Shell) (SSH) or [virtual private network](https://en.wikipedia.org/wiki/Virtual_private_network) (VPN).

**FTP over SSH**

FTP over SSH is the practice of tunneling a normal FTP session over a Secure Shell connection.[[23]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-nurdle-23) Because FTP uses multiple [TCP](https://en.wikipedia.org/wiki/Transmission_Control_Protocol) connections (unusual for a TCP/IP protocol that is still in use), it is particularly difficult to tunnel over SSH. With many SSH clients, attempting to set up a tunnel for the control channel (the initial client-to-server connection on port 21) will protect only that channel; when data is transferred, the FTP software at either end sets up new TCP connections (data channels) and thus have no [confidentiality](https://en.wikipedia.org/wiki/Confidentiality) or [integrity protection](https://en.wikipedia.org/wiki/Integrity_protection).

Otherwise, it is necessary for the SSH client software to have specific knowledge of the FTP protocol, to monitor and rewrite FTP control channel messages and autonomously open new [packet forwardings](https://en.wikipedia.org/wiki/Packet_forwarding) for FTP data channels. Software packages that support this mode include:

* Tectia ConnectSecure (Win/Linux/Unix)[[24]](https://en.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-24) of [SSH Communications Security](https://en.wikipedia.org/wiki/SSH_Communications_Security)'s software suite

**Derivatives**

**FTPS**

Main article: [FTPS](https://en.wikipedia.org/wiki/FTPS)

Explicit FTPS is an extension to the FTP standard that allows clients to request FTP sessions to be encrypted. This is done by sending the "AUTH TLS" command. The server has the option of allowing or denying connections that do not request TLS. This protocol extension is defined in [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [4217](https://tools.ietf.org/html/rfc4217). Implicit FTPS is an outdated standard for FTP that required the use of a SSL or TLS connection. It was specified to use different ports than plain FTP.

**SSH File Transfer Protocol**

Main article: [SSH File Transfer Protocol](https://en.wikipedia.org/wiki/SSH_File_Transfer_Protocol)

The SSH file transfer protocol (chronologically the second of the two protocols abbreviated SFTP) transfers files and has a similar command set for users, but uses the [Secure Shell](https://en.wikipedia.org/wiki/Secure_Shell) protocol (SSH) to transfer files. Unlike FTP, it encrypts both commands and data, preventing passwords and sensitive information from being transmitted openly over the network. It cannot interoperate with FTP software.

**Trivial File Transfer Protocol**

Main article: [Trivial File Transfer Protocol](https://en.wikipedia.org/wiki/Trivial_File_Transfer_Protocol)

Trivial File Transfer Protocol (TFTP) is a simple, lock-step FTP that allows a client to get a file from or put a file onto a remote host. One of its primary uses is in the early stages of [booting from a local area network](https://en.wikipedia.org/wiki/Network_booting), because TFTP is very simple to implement. TFTP lacks security and most of the advanced features offered by more robust file transfer protocols such as File Transfer Protocol. TFTP was first standardized in 1981 and the current specification for the protocol can be found in [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [1350](https://tools.ietf.org/html/rfc1350).

**Simple File Transfer Protocol**

Simple File Transfer Protocol (the first protocol abbreviated SFTP), as defined by [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [913](https://tools.ietf.org/html/rfc913), was proposed as an (unsecured) file transfer protocol with a level of complexity intermediate between TFTP and FTP. It was never widely accepted on the [Internet](https://en.wikipedia.org/wiki/Internet), and is now assigned Historic status by the [IETF](https://en.wikipedia.org/wiki/Internet_Engineering_Task_Force). It runs through port 115, and often receives the initialism of *SFTP*. It has a command set of 11 commands and support three types of data transmission: [ASCII](https://en.wikipedia.org/wiki/ASCII), [binary](https://en.wikipedia.org/wiki/Binary_file) and continuous. For systems with a [word size](https://en.wikipedia.org/wiki/Word_size) that is a multiple of 8 bits, the implementation of binary and continuous is the same. The protocol also supports login with user ID and password, hierarchical folders and file management (including *rename*, *delete*, *upload*, *download*, *download with overwrite*, and *download with append*).

**FTP commands**

Main article: [List of FTP commands](https://en.wikipedia.org/wiki/List_of_FTP_commands)

**FTP reply codes**

Main article: [List of FTP server return codes](https://en.wikipedia.org/wiki/List_of_FTP_server_return_codes)

Below is a summary of [FTP reply codes](https://en.wikipedia.org/wiki/List_of_FTP_server_return_codes) that may be returned by an FTP [server](https://en.wikipedia.org/wiki/Server_(computing)). These codes have been standardized in [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [959](https://tools.ietf.org/html/rfc959) by the IETF. The reply code is a three-digit value. The first digit is used to indicate one of three possible outcomes — success, failure, or to indicate an error or incomplete reply:

* 2yz – Success reply
* 4yz or 5yz – Failure reply
* 1yz or 3yz – Error or Incomplete reply

The second digit defines the kind of error:

* x0z – Syntax. These replies refer to syntax errors.
* x1z – Information. Replies to requests for information.
* x2z – Connections. Replies referring to the control and data connections.
* x3z – Authentication and accounting. Replies for the login process and accounting procedures.
* x4z – Not defined.
* x5z – File system. These replies relay status codes from the server file system.

The third digit of the reply code is used to provide additional detail for each of the categories defined by the second digit.

**See also**

* [Comparison of FTP client software](https://en.wikipedia.org/wiki/Comparison_of_FTP_client_software)
* [Comparison of FTP server software packages](https://en.wikipedia.org/wiki/Comparison_of_FTP_server_software_packages)
* [Comparison of file transfer protocols](https://en.wikipedia.org/wiki/Comparison_of_file_transfer_protocols)
* [Curl-loader](https://en.wikipedia.org/wiki/Curl-loader) – FTP/S loading/testing open-source software
* [File eXchange Protocol](https://en.wikipedia.org/wiki/File_eXchange_Protocol) (FXP)
* [File Service Protocol](https://en.wikipedia.org/wiki/File_Service_Protocol) (FSP)
* [FTAM](https://en.wikipedia.org/wiki/FTAM)
* [FTPFS](https://en.wikipedia.org/wiki/FTPFS)
* [List of FTP commands](https://en.wikipedia.org/wiki/List_of_FTP_commands)
* [List of FTP server return codes](https://en.wikipedia.org/wiki/List_of_FTP_server_return_codes)
* [Managed File Transfer](https://en.wikipedia.org/wiki/Managed_File_Transfer)
* [OBEX](https://en.wikipedia.org/wiki/OBEX)
* [Shared file access](https://en.wikipedia.org/wiki/Shared_file_access)
* [TCP Wrapper](https://en.wikipedia.org/wiki/TCP_Wrapper)

**References**

 *Forouzan, B.A. (2000). TCP/IP: Protocol Suite (1st ed.). New Delhi, India: Tata McGraw-Hill Publishing Company Limited.*

  *Kozierok, Charles M. (2005).* [*"The TCP/IP Guide v3.0"*](http://www.tcpipguide.com/free/t_FTPOverviewHistoryandStandards.htm)*. Tcpipguide.com.*

  *Dean, Tamara (2010). Network+ Guide to Networks. Delmar. pp. 168–171.*

  *Clark, M.P. (2003). Data Networks IP and the Internet (1st ed.). West Sussex, England: John Wiley & Sons Ltd.*

  [*"Active FTP vs. Passive FTP, a Definitive Explanation"*](http://slacksite.com/other/ftp.html)*. Slacksite.com.*

  [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [959](https://tools.ietf.org/html/rfc959) (Standard) File Transfer Protocol (FTP). Postel, J. & Reynolds, J. (October 1985).

  [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [2428](https://tools.ietf.org/html/rfc2428) (Proposed Standard) Extensions for IPv6, NAT, and Extended Passive Mode. Allman, M. & Metz, C. & Ostermann, S. (September 1998).

  *Stevens, W. Richard (1994). TCP/IP Illustrated Volume I.* ***1****. Reading, Massachusetts, USA: Addison-Wesley Publishing Company.* [*ISBN*](https://en.wikipedia.org/wiki/ISBN_(identifier))[*0-201-63346-9*](https://en.wikipedia.org/wiki/Special:BookSources/0-201-63346-9)*.*

  *Gleason, Mike (2005).* [*"The File Transfer Protocol and Your Firewall/NAT"*](https://www.ncftp.com/ncftpd/doc/misc/ftp_and_firewalls.html)*. Ncftp.com.*

  *Klensin, John.* [*"FTP TYPE Extension for Internationalized Text"*](https://tools.ietf.org/html/draft-klensin-ftpext-typeu-00.html)*. tools.ietf.org. Retrieved 9 June 2020.*

  *Preston, J. (January 2005).* [*Deflate transmission mode for FTP*](https://tools.ietf.org/html/draft-preston-ftpext-deflate-03)*.* [*IETF*](https://en.wikipedia.org/wiki/Internet_Engineering_Task_Force)*. I-D draft-preston-ftpext-deflate-03. Retrieved 27 January 2016.*

  *Allcock, W. (April 2003).* [*"GridFTP: Protocol Extensions to FTP for the Grid"*](https://www.ogf.org/documents/GFD.20.pdf) *(PDF).*

  *Mandrichenko, I. (4 May 2005).* [*"GridFTP v2 Protocol Description"*](https://www.ogf.org/documents/GFD.47.pdf) *(PDF).*

  *Prince, Brian.* [*"Should Organizations Retire FTP for Security?"*](https://www.securityweek.com/should-organizations-retire-ftp-security)*. Security Week. Security Week. Retrieved 14 September 2017.*

  [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [1635](https://tools.ietf.org/html/rfc1635) (Informational) How to Use Anonymous FTP. P. & Emtage, A. & Marine, A. (May 1994).

  *Matthews, J. (2005). Computer Networking: Internet Protocols in Action (1st ed.). Danvers, MA: John Wiley & Sons Inc.*

  *Abrams, Lawrence (26 November 2018).* [*"Chrome and Firefox Developers Aim to Remove Support for FTP"*](https://www.bleepingcomputer.com/news/google/chrome-and-firefox-developers-aim-to-remove-support-for-ftp/)*. bleepingcomputer.com. Retrieved 26 January 2020.*

  [*"1574475 - Remove FTP support"*](https://bugzilla.mozilla.org/show_bug.cgi?id=1574475)*.*

  [*"Deprecate FTP support - Chrome Platform Status"*](https://www.chromestatus.com/feature/6246151319715840)*.*

  [*"Accessing FTP servers | How to | Firefox Help"*](https://support.mozilla.org/en-US/kb/Accessing%20FTP%20servers#w_ftp-servers-that-require-a-username-and-password)*. Support.mozilla.com. 5 September 2012. Retrieved 16 January 2013.*

  [How to Enter FTP Site Password in Internet Explorer](https://web.archive.org/web/20150702005840/https:/support.microsoft.com/en-us/kb/135975) at the [Wayback Machine](https://en.wikipedia.org/wiki/Wayback_Machine) (archived 2015-07-02) Written for IE versions 6 and earlier. Might work with newer versions.

  *Jukka “Yucca” Korpela (18 September 1997).* [*"FTP URLs"*](http://jkorpela.fi/ftpurl.html)*. "IT and communication" (jkorpela.fi). Retrieved 26 January 2020.*

  [*"Securing FTP using SSH"*](https://nurdletech.com/linux-notes/ftp/ssh.html)*. Nurdletech.com.*

* 1.  [*"Components of the Information Assurance Platform (section Tectia ConnectSecure)"*](https://www.ssh.com/manuals/mft-events-product/63/ssh-solutions-your-business-components.html)*. ssh.com.*

**Further reading**

* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [697](https://tools.ietf.org/html/rfc697) – CWD Command of FTP. July 1975.
* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [959](https://tools.ietf.org/html/rfc959) – (Standard) File Transfer Protocol (FTP). J. Postel, J. Reynolds. October 1985.
* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [1579](https://tools.ietf.org/html/rfc1579) – (Informational) Firewall-Friendly FTP. February 1994.
* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [1635](https://tools.ietf.org/html/rfc1635) – (Informational) How to Use Anonymous FTP. May 1994.
* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [1639](https://tools.ietf.org/html/rfc1639) – FTP Operation Over Big Address Records (FOOBAR). June 1994.
* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [1738](https://tools.ietf.org/html/rfc1738) – Uniform Resource Locators (URL). December 1994.
* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [2228](https://tools.ietf.org/html/rfc2228) – (Proposed Standard) FTP Security Extensions. October 1997.
* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [2389](https://tools.ietf.org/html/rfc2389) – (Proposed Standard) Feature negotiation mechanism for the File Transfer Protocol. August 1998.
* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [2428](https://tools.ietf.org/html/rfc2428) – (Proposed Standard) Extensions for IPv6, NAT, and Extended passive mode. September 1998.
* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [2577](https://tools.ietf.org/html/rfc2577) – (Informational) FTP Security Considerations. May 1999.
* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [2640](https://tools.ietf.org/html/rfc2640) – (Proposed Standard) Internationalization of the File Transfer Protocol. July 1999.
* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [3659](https://tools.ietf.org/html/rfc3659) – (Proposed Standard) Extensions to FTP. P. Hethmon. March 2007.
* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [5797](https://tools.ietf.org/html/rfc5797) – (Proposed Standard) FTP Command and Extension Registry. March 2010.
* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [7151](https://tools.ietf.org/html/rfc7151) – (Proposed Standard) File Transfer Protocol HOST Command for Virtual Hosts. March 2014.
* [IANA FTP Commands and Extensions registry](https://www.iana.org/assignments/ftp-commands-extensions/ftp-commands-extensions.xhtml) – The official registry of FTP Commands and Extensions

**External links**

 [](https://en.wikipedia.org/wiki/File:Wikibooks-logo-en-noslogan.svg)[Communication Networks/File Transfer Protocol](https://en.wikibooks.org/wiki/Communication_Networks/File_Transfer_Protocol) at Wikibooks

 [FTP Server Online Tester](https://servertest.online/ftp) Authentication, encryption, mode and connectivity.

 Anonymous FTP Servers by Country Code [TLD](https://en.wikipedia.org/wiki/Top-level_domain) (2012): [*"Offbeat Internet - Public Access - FTP"*](https://www.jumpjet.info/Offbeat-Internet/Public/FTP/url.htm)*. www.jumpjet.info. 2012. Retrieved 16 January 2020.*