**ftp Library Package 2.1 for Tcl/Tk Manual Pages**

**COMMAND** **ftp::Open**  *server  user  passwd  ?options?*   The **ftp::Open** command is used to start the FTP session by establishing a control connection to the FTP server. If no options are specified, then the defaults are used.

The **ftp::Open** command takes a host name *server*, a user name *user* and a password *password* as its parameters and returns a session handle that is an integer greater than or equal to 0 if the connection is successfully established, otherwise it returns "-1".

The *server* parameter must be the name or internet address (in dotted decimal notation) of the ftp server. The *user* and *passwd* parameters must contain a valid user name and password to complete the login process.

The options overwrite some default values or set special abilities:

**-blocksize size**

The blocksize is used during data transfer. At most *size* bytes are transfered at once. After each block, a call to the "-progress callback" is made. The default value for this option is 4096.

**-timeout seconds**

If *seconds* is non-zero, then **ftp::Open** sets up a timeout to occur after the specified number of seconds. The default value is 600.

**-port number**

The *port number* specifies an alternative remote port on the ftp server on which the ftp service resides. Most ftp services listen for connection requests on default port 21. Sometimes, usually for security reasons, port numbers other than 21 are used for ftp connections.

**-mode mode**

The *transfer mode* option determines if a file transfer occurs in an active or passive way. In passive mode the client session may want to request the ftp Server to listen for a data port and wait for the connection rather than initiate the process when a data transfer request comes in. Passive mode is normally a requirement when accessing sites via a firewall. The default mode is active.

**-progress callback**

The *callback* is made after each transfer of a data block specified in blocksize. The callback gets as additional argument the current number of bytes transferred so far. Here is a template for the progress callback:

proc Progress {total} {  
 puts "$total bytes transfered!"  
}

**-command callback**

Specifying this option puts the connection in asynchronous mode. The *callback* is made after each operation has been completed. The callback gets as an additional argument a keyword of the operation that has completed plus additional arguments specific to the operation. If an error occurs the callback is made with the keyword "error". When an operation, such as "Cd", "Get", and so on, has been started no further operations should be started until a callback has been received for the current operation. A template for the callback is:

proc Callback {what args} {  
 puts "Operation $what $args completed"  
}

**EXAMPLE**

set server "ftp.server.com"  
set user "anonymous"  
set passwd "mist@foo.com"  
  
# define callback  
proc Progress {total} {  
 puts "$total bytes transfered!"  
}  
  
# open a new connection  
if {[set conn [ftp::Open $server $user $passwd -progress Progress -blocksize 1024 -mode passive]] == -1} {  
 puts "Connection refused!"  
 exit 1  
}  
  
# get a file  
ftp::Get $conn index.html  
  
# close connection  
ftp::Close $conn

[[Contents](http://docs.google.com/index.html)]  [**Next:** [ftp::Close](http://docs.google.com/fhelp2.html)]

© 1999 [Steffen Traeger](mailto:Steffen.Traeger@t-online.de)