



## Basic Usage

### Connecting to Servers

<code>ncftp</code>	Starts ncftp in interactive mode.
<code>ncftp &lt;hostname&gt;</code>	Connects to the specified hostname using anonymous login.
<code>ncftp -u &lt;username&gt; &lt;hostname&gt;</code>	Connects to the specified hostname using the provided username.
<code>ncftp -u &lt;username&gt;:&lt;password&gt; &lt;hostname&gt;</code>	Connects using both username and password directly (less secure, use with caution).
<code>open &lt;hostname&gt;</code>	Connects to the specified hostname within the ncftp interactive mode.
<code>open -u &lt;username&gt; &lt;hostname&gt;</code>	Connects using a specific username within interactive mode, prompting for a password.

### Basic File Operations

<code>ls</code>	Lists files in the current remote directory.
<code>dir</code>	Same as <code>ls</code> , lists files in the current remote directory.
<code>pwd</code>	Prints the current remote directory.
<code>cd &lt;directory&gt;</code>	Changes the current remote directory.
<code>lcd &lt;directory&gt;</code>	Changes the current local directory.
<code>get &lt;remote_file&gt; [local_file]</code>	Downloads a file from the remote server. Optionally, save it with a different name locally.

### Transferring Files

<code>get &lt;remote_file&gt; [local_file]</code>	Downloads a file from the remote server. If <code>local_file</code> is not specified, it uses the same name.
<code>put &lt;local_file&gt; [remote_file]</code>	Uploads a file to the remote server. If <code>remote_file</code> is not specified, it uses the same name.
<code>mget &lt;remote_files&gt;</code>	Downloads multiple files from the remote server using wildcards.
<code>mput &lt;local_files&gt;</code>	Uploads multiple files to the remote server using wildcards.
<code>ncftpget -R &lt;bookmarkname&gt; &lt;remote_dir&gt; &lt;local_dir&gt;</code>	Recursively downloads an entire directory from the server specified by a bookmark.
<code>ncftpput -R &lt;bookmarkname&gt; &lt;local_dir&gt; &lt;remote_dir&gt;</code>	Recursively uploads an entire directory to the server specified by a bookmark.

## Advanced Features

### Background Processing

<code>bgget &lt;remote_file&gt; [local_file]</code>	Downloads a file in the background.
<code>bgput &lt;local_file&gt; [remote_file]</code>	Uploads a file in the background.
<code>bkmkadd &lt;bookmarkname&gt;</code>	Adds a bookmark for the current connection.
<code>bkmk &lt;bookmarkname&gt;</code>	Connects to a bookmarked site.
<code>jobs</code>	Lists background jobs.
<code>kill &lt;jobnumber&gt;</code>	Kills a background job.

### Bookmarks

<code>bookmark &lt;bookmarkname&gt;</code>	Connects to a bookmarked server.
<code>bkmkadd &lt;bookmarkname&gt;</code>	Adds a bookmark for the current connection.
<code>bkmkdel &lt;bookmarkname&gt;</code>	Deletes a bookmark.
<code>bkmklist</code>	Lists all bookmarks.
<code>open -b &lt;bookmarkname&gt;</code>	Opens a connection using the specified bookmark.
<code>bookmark bookmarkname</code>	Bookmarks the current FTP location.

Resuming Transfers

<code>get -z &lt;remote_file&gt; [local_file]</code>	Attempts to resume a download.
<code>put -z &lt;local_file&gt; [remote_file]</code>	Attempts to resume an upload.
<code>ncftpget -C &lt;bookmark&gt; &lt;remote_file&gt; &lt;local_file&gt;</code>	Resume transfer based on a bookmark and specific files.
<code>ncftpput -C &lt;bookmark&gt; &lt;local_file&gt; &lt;remote_file&gt;</code>	Resume upload, specifying bookmarks and files.

Command Line Options

General Options

<code>ncftp -d</code>	Enables debugging mode.
<code>ncftp -v</code>	Verbose mode, shows more details.
<code>ncftp -T &lt;seconds&gt;</code>	Sets transfer timeout.
<code>ncftp -u &lt;username&gt;</code>	Specifies the username for login.
<code>ncftp -p &lt;port&gt;</code>	Specifies the port number to connect to.
<code>ncftp -o &lt;logfile&gt;</code>	Outputs ncftp's log messages to the specified file.

Transfer Options

<code>ncftp -C</code>	Continues or resumes interrupted transfers.
<code>ncftp -z</code>	Tries to use the FXP protocol (server-to-server transfer).
<code>ncftp -DD</code>	Delete the source files after successful transfer.
<code>ncftp -E</code>	Force use of extended passive mode (EPSV).
<code>ncftp -F</code>	Force use of passive mode (PASV).
<code>ncftp -a</code>	Use ASCII transfer mode.

ncftpget and ncftpput Specific Options

<code>ncftpget -R</code>	Recursive retrieval of directories.
<code>ncftpput -R</code>	Recursive uploading of directories.
<code>ncftpget -T &lt;secs&gt;</code>	Transfer timeout in seconds.
<code>ncftpput -S &lt;secs&gt;</code>	Script timeout in seconds.
<code>ncftpget -dd</code>	Delete the source file after successfully downloading.
<code>ncftpput -zz</code>	Create missing directories.

Tips and Tricks

Efficient Transfers

Use bookmarks for frequently accessed servers to save time and avoid retyping credentials.  <code>bkmkadd myserver</code> <code>open -b myserver</code>
For large directory transfers, use the <code>-R</code> option with <code>ncftpget</code> and <code>ncftpput</code> for recursive operations.  <code>ncftpget -R myserver /remote/path /local/path</code>
Leverage background transfers ( <code>bgget</code> , <code>bgput</code> ) to continue working while files are transferred.  <code>bgget large_file.zip</code> <code>jobs</code>
Utilize the <code>-C</code> option to resume interrupted transfers, ensuring data integrity and saving bandwidth.  <code>get -C interrupted_file.iso</code>
Set transfer and script timeouts ( <code>-T</code> , <code>-S</code> ) to prevent stalled connections from blocking the client.  <code>ncftp -T 30 server</code>

Troubleshooting Common Issues

If experiencing connection problems, ensure the firewall allows FTP traffic on port 21 (control) and ports for data transfer.  <code>Check firewall settings.</code>
When transfers fail, verify that the remote directory exists and that you have the necessary permissions.  <code>Use 'ls -l' to check permissions.</code>
If passive mode is required, use the <code>-F</code> option or set <code>use-pasv</code> to <code>yes</code> in <code>~/.ncftp/config</code> .  <code>Set 'use-pasv yes' in config.</code>
When encountering timeout issues, increase the transfer timeout using the <code>-T</code> option.  <code>ncftp -T 60 server</code>
To resolve issues with ASCII vs. binary transfers, force the transfer mode using <code>-a</code> (ASCII) or <code>-b</code> (binary).  <code>get -b binary_file</code>

Automation with Scripts

ncftp can be used in scripts for automated file transfers. Here's a basic example:  <pre>#!/bin/bash ncftp -u user:password ftp.example.com &lt;&lt;EOF cd /remote/directory get important_file.txt /local/path bye EOF</pre>
This script connects to an FTP server, navigates to a remote directory, downloads a file, and then exits.
For more complex tasks, you can chain commands using the <code>-e</code> option, executing commands sequentially. However, this is generally less reliable than using a script.  <code>ncftp -u user:pass -e "cd /path; get file.txt local.txt; bye" host.com</code>
When automating, ensure proper error handling. Use <code>-S</code> to set a script timeout, preventing indefinite hangs in case of errors.  <code>ncftpput -S 60 -R bookmark /local/dir /remote/dir</code>