



# Fetch Help

[fetchsoftworks.com](http://fetchsoftworks.com)

[About Fetch](#)

[What's New](#)

[Quick Start](#)

[Using Fetch](#)

[Troubleshooting](#)

[Reference](#)

[Glossary](#)



## [Fetch Help](#) > **About**

Fetch is an easy-to-use, full-featured file transfer (FTP and SFTP) client for the Apple Macintosh.

Select a topic below for more information about Fetch.

[About Fetch](#)

[Contact Information](#)

[Licensing & Purchasing](#)

[Upgrading from an Earlier Version of Fetch](#)

[Uninstalling Fetch](#)

[Credits and Acknowledgments](#)



[Fetch Help](#) > [About](#) > **About Fetch**

Fetch is a Macintosh program that makes it easy to transfer files between your Macintosh and other computers over networks such as the Internet. Fetch transfers files using the File Transfer Protocol (FTP) or SSH File Transfer Protocol (SFTP).

Fetch can be used to:

- Publish a website at a web hosting provider.
- Rename, move, delete, and change the permissions of files on a web server in order to maintain a website.
- Transfer documents to a printing company or service bureau.
- Submit advertisements to newspapers and magazines.
- Publish images for eBay auction listings.
- Move files between your Macintosh and Macintosh, UNIX, Windows, NetWare, or other sorts of computers, as well as specialized equipment including medical imaging machines, image editing systems, and photo processing equipment.

Fetch 5.6 runs on Mac OS X 10.4 or later.

For more details, see the [What is Fetch?](#) help topic.

For information about licensing and purchasing Fetch, see the [Licensing Fetch](#) help topic.

- [Fetch Credits and Acknowledgments](#)
- [Fetch Softworks Contact Information](#)



[Fetch Help](#) > [About](#) > **Contacting Fetch Softworks**

### **Technical support**

Please see the [Troubleshooting & Support](#) help topic for suggestions to try before contacting support.

Message boards: <http://fetchsoftworks.com/messageboard>

Email: [support@fetchsoftworks.com](mailto:support@fetchsoftworks.com)

### **Company contact information**

Website: <http://fetchsoftworks.com>

Email: [info@fetchsoftworks.com](mailto:info@fetchsoftworks.com)

Fetch Softworks  
PO Box 72  
Etna, NH 03750  
USA

Voicemail/FAX: +1 603-619-0397

 [Fetch Help](#) > [About](#) > **Licensing Fetch**

A single-user license for Fetch 5.6 is \$29 US. You can purchase a license by choosing [Purchase](#) from the **Fetch** menu. Users at educational or non-profit charitable organizations (government research laboratories excepted) may [apply](#) for a free license. Multi-user licenses are also available. For more information, visit our [website](#).

Upgrades to Fetch 5.6 are free for registered users of Fetch 5.5 and later, or if you purchased Fetch after January 28, 2009. Otherwise, registered users of Fetch 5.3.1 and earlier can upgrade for \$10. To upgrade, choose [Purchase Upgrade](#) from the **Fetch** menu.

You can try Fetch for free for 15 days. Fetch is fully functional during the trial period.

If your **Fetch** menu does not have a **Purchase** or **Purchase Upgrade** menu item, your copy of Fetch already has a valid license. Choose **About Fetch** from the **Fetch** menu to view the license serial number. Visit the [Fetch Softworks website](#) if you wish to purchase additional copies.



## [Fetch Help](#) > [About](#) > **Upgrading from an Earlier Version of Fetch**

If you're currently using an earlier version of Fetch, upgrading to Fetch 5.6 is easy. Fetch 5.6 will automatically read your old Fetch preferences and shortcuts when you first open it, and any saved Fetch documents will work with Fetch 5.6. You can continue to keep and use earlier versions of Fetch on your hard drive if you want.

For information on the new features and changes in Fetch 5.6, see:

- [What's New](#)
- [Fetch 5.6 for Fetch 5 Users](#)
- [Fetch 5.6 for Fetch 4 Users](#)

Registered users of any previous version of Fetch 5 or Fetch 4 are eligible for a discount when purchasing Fetch 5.6. See the [Licensing Fetch](#) help topic for information on pricing and purchasing an upgrade.



## [Fetch Help](#) > [About](#) > **Uninstalling Fetch**

If you want to completely uninstall Fetch from your Macintosh, you need to remove the Fetch application and the Fetch widget.

To uninstall the Fetch application:

1. In the Finder, choose **Go > Applications** to open the Applications folder.
2. Locate the Fetch icon.
3. Drag the Fetch icon to the Trash.

You can uninstall the [Fetch Dashboard widget](#) using the Manage Widgets feature of Dashboard:

1. Open Dashboard, either by clicking on Dashboard icon in the Dock or by pressing the Dashboard keyboard shortcut (usually the F12 key).
2. Click on the Open (+) button in the bottom-left of the screen to reveal the widget bar.
3. Click Manage Widgets to open the Widgets Dashboard widget.
4. Scroll the list of widgets until you have located the Fetch widget.
5. Click on the red Remove (-) button after the name of the Fetch widget.
6. Click OK to confirm that you want to move the widget to the Trash.

The Fetch Automator actions will be uninstalled when you uninstall the Fetch application.

Fetch also creates several preferences files in the Preferences folder of the Library folder in your home folder. While it does no harm to leave these files on your computer after you uninstall Fetch, for completeness, these files are named:

```
com.fetchsoftworks.Fetch.plist  
com.fetchsoftworks.Fetch.Shortcuts.plist  
Fetch Cache  
widget-com.fetchsoftworks.widget.fetchupload.plist
```

Older versions of Fetch may have also created preferences files named:

```
Fetch Prefs  
Fetch Shortcuts
```



## [Fetch Help](#) > [About](#) > Credits and Acknowledgments

Engineering by Jim Matthews, Ben Artin, and Scott McGuire.

Testing by Doug Grinbergs and Scott McGuire.

Documentation by Scott McGuire.

Copyright © 2001–2010, Fetch Softworks.

Copyright © 2000, Trustees of Dartmouth College.

Fetch is a product of Fetch Softworks.

Uses RSA Data Security, Inc. MD4 & MD5 message digest algorithms.

Other product and company names mentioned herein may be trademarks or trade names of their respective holders.

Thanks to:

- Anthony Piraino of The Iconfactory for icons, art, and the new running dog.
- Ron Avitzur, Rob Barris, Marshall Clow, Tim Dierks, Chris Espinosa, Mat Marcus, John Penn, Quinn 'The Eskimo,' Jonathan 'Wolf' Rentzsch, Leonard Rosenthal, Eric Schlegel, Rich Siegel, Keith Stattenfield, James Thomson, and Russell Williams, for assorted technical assistance.
- John Norstad and Northwestern University, for resource file checking code.
- Wordstation and Jean–Pierre Kuypers for the localizations.
- Tadashi Suzuki and Hart Computer for the logo.
- Kate Fagerstrom for the new disk image background and other graphics.
- Robin Olson for the previous disk image background and widget graphics.
- Mitch Jones of Allume for the copious help with Stuffit.
- Kathy Kimball for slogans and ad design.
- Cathy Richardson and Chapin for the bark.
- Apple's Human Interface Group for helpful advice.
- Dartmouth College for supporting Fetch's development 1989–2000.
- All the users who reported problems and suggested new features.
- All the individuals and companies who licensed version 4.
- Margie, Daniel, and Dylan, for their love and support.



## [Fetch Help](#) > What's New in Fetch 5.6

Fetch 5.6 adds automatic resume of stalled or failed downloads, and improves the behavior of the automatic resume of uploads in various circumstances. This is in addition to the new features introduced in Fetch 5.5's, which are the ability to edit any kind of file using any application and the changes will automatically be saved back to the server, automatic resume of stalled or failed uploads, automatic resume of stalled or failed downloads, and Quick Look support for previewing files on servers.

If you're a regular user of Fetch upgrading from an earlier version, see either the [Fetch 5.6 for Fetch 5 Users](#) or the [Fetch 5.6 for Fetch 4 Users](#) guide to learn about new features and changes.

See the [summary of new features in Fetch 5.6](#) topic for a simple list of the major changes and improvements, or read the complete list of the changes, new features, and fixes in the [Fetch 5.6 release notes](#).

AppleScript users should read the [AppleScript changes](#) help topic for information about changes to Fetch's AppleScript terminology.

The entire Fetch Help is available as a [downloadable PDF file](#) from the Fetch Softworks website. It serves as the Fetch manual.

This is just a summary; for a more extensive list of changes and improvements, see the [Fetch 5.6 Release Notes](#).

The major new features of Fetch 5.6 are:

- Added automatic resume of stalled or failed downloads
- Improved automatic resume of uploads in various circumstances

The major new features of Fetch 5.5 were:

- Improved [Edit](#) command — you can now edit any kind of file using any application and the changes will automatically be saved back to the server
- Added automatic resume of stalled or failed uploads
- Added [Quick Look](#) support for previewing files on servers (Mac OS X 10.5 and later only)
- Added support for gestures on laptop trackpads: in the transfer window, swipe left to go back to previously viewed folder, swipe up to go to the parent folder, and swipe down to open the selected folder
- Made Get Info quicker by not calculating folder sizes until you ask for them

The major new features of Fetch 5.3 were:

- Improved compatibility with Mac OS X 10.5 Leopard
- Improved transfer window appearance on Mac OS X 10.5
- Redesigned toolbar and button icons with updated look for Mac OS X 10.5
- Added preference to [turn off automatic decoding](#) for all file types
- Digitally signed Fetch application, to reduce unnecessary keychain alerts
- Added support for Mac OS X 10.5's [application-specific firewall](#)
- Added support for [uploading files using Copy and Paste](#)
- Added support for [copying files on servers using Copy and Paste](#)

The major new features of Fetch 5.2 were:

- Added [WebView](#) — easily view files in your web browser and copy web addresses while in Fetch
- Added support for [FTP with TLS/SSL \(FTPS\)](#) connections
- Added [droplet shortcuts](#) — drag files to a droplet shortcut's icon to upload them to your server
- Overhauled FTP networking for improved compatibility and performance
- Moved [Refresh](#) button from an optional toolbar button to a permanent button on transfer window info bar
- Added [Get As](#) command that prompts you to choose a folder in which to save downloads

The major new features of Fetch 5.1 were:

- Added support for Intel-based Macintosh computers — Fetch is now a Universal binary
- Added [Fetch Dashboard widget](#), for easy uploading and checking on the progress of your transfer (the widget is installed automatically when you start Fetch)
- Added [Automator actions](#) for Fetch, including Upload Files, Download Files, Set Fetch Item Permissions, Move Fetch Items, Rename Fetch Items, and more. Read the [introduction](#) to and [examples](#) of using Automator with Fetch
- Made the New Connection dialog modeless — you can now use other Fetch windows

while the New Connection dialog is open

- Included StuffIt Engine with Fetch application — separate installation or updating of StuffIt is no longer required to use StuffIt support in Fetch
- [Zip Archive format](#) now preserves Macintosh information (Fetch creates .zip files compatible with the Finder's Compress/Create Archive feature)

The major new features of Fetch 5 were:

- Added support for transferring files using [SFTP](#) (SSH File Transfer Protocol)
- Redesigned the file transfer interface to include:
  - New [toolbar](#) for easy access to common Fetch functions
  - Back and Recent folders buttons for easier navigation
  - [File list](#) that works more like the Finder list view
  - Compact [status pane](#) at the bottom of transfer window
- Improved the Fetch user experience on Mac OS X
- Added Dutch, French, German, Italian, Spanish, and Swedish localizations
- Reworded error messages to use everyday language
- Reorganized [menus](#) for clarity
- Added context-sensitive help, including help tags and extensive Apple Help book
- Made Fetch Help available as a [downloadable PDF](#)
- Added Automatic Passive Mode — automatic detection of proper connection mode for more reliable transfers through firewalls and NAT devices
- Added support for [uploading files as StuffIt X archives](#), Zip archives, Gzip, or Gzipped Tar archives
- Added support for [automatically decoding](#) StuffIt X, Zip, Gzip, Gzipped Tar, and many more formats when downloading
- Added support for [resuming uploads](#) of binary file transfers
- Added [Bonjour](#) support — Fetch will automatically detect and list FTP or SFTP servers available via Bonjour (formerly called Rendezvous)
- Made file transfers faster
- Added a file transfer progress indicator to the Dock icon
- Added a recent connections pop-up menu to the [New Connection dialog](#)
- Added [automatic checks for updates](#) to Fetch, with easy update downloading
- Added an easier, more powerful interface for [setting file types, formats, and transfer options](#)
- Added support for in-place [renaming of files](#) and editing shortcut information
- Added support for importing [Transmit favorites](#) and [Interarchy bookmarks](#)
- Improved compatibility with non-English servers
- Improved support for [non-ASCII and Unicode filenames](#)
- Improved compatibility with Windows, VMS, VM/CMS, VSFTPD, FileZilla, and WFTPD servers



This is an extensive list of all the changes and improvements in Fetch 5.6. For a shorter summary of the biggest improvements, see the [summary of new features in Fetch 5.6](#).

[Fetch 5.6 Release Notes](#)

[Fetch 5.5.3 Release Notes](#)

[Fetch 5.5.2 Release Notes](#)

[Fetch 5.5.1 Release Notes](#)

[Fetch 5.5 Release Notes](#)

[Fetch 5.3.1 Release Notes](#)

[Fetch 5.3 Release Notes](#)

[Fetch 5.2.1 Release Notes](#)

[Fetch 5.2 Release Notes](#)

[Fetch 5.1.1 Release Notes](#)

[Fetch 5.1 Release Notes](#)

[Fetch 5.0.5 Release Notes](#)

[Fetch 5.0.4 Release Notes](#)

[Fetch 5.0.3 Release Notes](#)

[Fetch 5.0.2 Release Notes](#)

[Fetch 5.0.1 Release Notes](#)

[Fetch 5 Release Notes](#)

## **Fetch 5.6**

- Added automatic resume of stalled or failed downloads
- Improved stalled transfer detection to work better in more circumstances
- Fixed some cases where automatically resumed uploads would try to resume when the entire file was already on the server
- Improved handling of uploads where the connection is dropped after an hour with certain routers, to avoid unnecessary resume upload attempts
- Made detection and automatic resume of FTP upload stalls quicker
- Added automatic resume of SFTP text files uploads in most cases (when not converting line endings)
- Improved behavior of automatic upload resume on very slow connections
- Fixed a problem where Adobe Flash (.swf) files would sometimes be uploaded in the incorrect format
- Fixed a problem where the Fetch widget would not dismiss Dashboard on Mac OS X 10.6 and later
- Improved scrolling performance in folders containing hundreds of different file kinds
- Changed system requirements to Mac OS X 10.4 or later
- Improved encoding handling when editing files with BBEdit or TextWrangler; if a file specifies an encoding, it will be used instead of Fetch's preferred encoding preference
- Added server-side serial number verification
- Changed droplet shortcuts to queue uploads if an upload to the same server is already in progress, instead of opening a new transfer window
- Changed "immediate return" AppleScript property to queue uploads if an upload to the same server is already in progress, instead of opening a new transfer window
- Fixed a rare problem where certain kinds of font files were uploaded as zero length files
- Added connection error messages to Fetch Transcript
- Improved descriptions of transfer results in Fetch Transcript
- Fixed a problem with New Connection dialog moving to background after Option-double-clicking a shortcut

- Fixed Automatic Passive Mode to immediately honor changes to "Use passive mode transfers" preference
- Fixed a problem where "Automatically decode files like this" setting for zip files was not remembered on Mac OS X 10.4
- Changed preferences to default to not sending MACB probe
- Fixed incorrect upload format display in Put button help tag
- Fixed missing "Bounce icon in Dock" notifications
- Revised unknown item icon
- Fixed rare -108 error after upgrading from Fetch 4.0.3
- Improved error message for POSIX Error #51

### Fetch 5.5.3

- Fixed Mac OS -8905 errors when opening Fetch or saving shortcuts on Mac OS X 10.6 Snow Leopard
- Fixed rare freezes when quitting
- Fixed a problem where Fetch displayed the copy cursor when moving a file
- Fixed problem displaying file lists with Norwegian month names from vsFTPd servers
- Fixed a problem where items moved to a parent folder would sometimes not appear in the parent folder's file list
- Improved compatibility with certain versions of the VanDyke VShell server that send non-standard TYPE replies
- Fixed repeated server messages in transcript window
- Changed handling of download stalls — Fetch now displays an error instead of waiting forever

### Fetch 5.5.2

- Improved Quick Look behavior when using Mac OS X 10.6 and later — Quick Look windows now include Full Screen and Add to iPhoto buttons, and you can switch back and forth between the Index Sheet and individual images
- Replaced Quick Look as Text with more flexible [View as Text](#) command — View as Text allows you to open multiple windows, cut and paste text, and view the windows while Fetch is in the background
- Added support for SFTP through SOCKS proxies
- Fixed problem adding Get Specific Fetch Items action to Automator workflows on Mac OS X 10.6
- Fixed a problem in the Purchase window where editing the email address would sometimes change the country field
- Disabled Stop button in Purchase window when stopping the purchase may not be possible
- Fixed problems loading widget on certain systems that have StuffIt Deluxe installed
- Fixed problems automatically resuming uploads on certain Filezilla servers
- Fixed problems automatically resuming uploads on certain WS\_FTP servers
- Fixed a problem duplicating files on certain servers
- Fixed a problem where the editor choice sometimes would not be remembered when the "Remember this editor choice" checkbox was checked
- Fixed a problem using Edit New with JavaScript files when Dashcode is the editor
- Improved compatibility of automatic passive mode with certain servers
- Improved compatibility with Chyron character generators

### Fetch 5.5.1

- Fixed rare crashes when uploading caused by missing default editors

- Fixed rare crashes when opening preferences caused by missing default editors
- Fixed rare freezes when getting file lists
- Fixed delays in renaming items when the list of recent folders is very long
- Fixed a problem where pressing Return in the Put dialog inserted a space into the names of uploaded files
- Made it easier to select all files in the Put dialog when a single file is already selected
- Fixed a problem where file lists wouldn't be displayed when connecting to certain Windows servers and the "Use LIST -al" preference was selected
- Fixed a problem where folders would display as files when connected to certain Microsoft DFS servers
- Fixed a problem downloading folders that appear to contain files beginning with ".\_" from certain servers
- Fixed garbled text in confirmation dialog when dragging files to the Trash
- Fixed rare problem where Fetch Shortcuts window could become very short and not resizable

## Fetch 5.5

The major new features in Fetch 5.5 are:

- Improved [Edit](#) command — you can now edit any kind of file using any application and the changes will automatically be saved back to the server
- Added automatic resume of stalled or failed uploads
- Added [Quick Look](#) support for previewing files on servers (Mac OS X 10.5 and later only)
- Added support for gestures on laptop trackpads: in the transfer window, swipe left to go back to previously viewed folder, swipe up to go to the parent folder, and swipe down to open the selected folder
- Made Get Info quicker by not calculating folder sizes until you ask for them

Other fixes and improvements in Fetch 5.5 are:

- Fixed missing modification dates in Put, Open, and Save dialogs on Mac OS X 10.5 and later
- Changed transfer window title bar and Recent Connections menu to show shortcut name instead of hostname when there's a matching shortcut
- Changed transfer window title bar to list folder name first, followed by shortcut name or hostname
- Changed file list to remember the last sorting choice
- Redesigned and streamlined purchase window
- Added Edit With menu for quickly editing a file with a non-default editor
- Improved Edit With command to let you to choose any compatible editing application
- Added "Remember this editor choice" option to Edit With dialog to permanently set a new editor for the selected kind of file
- Fixed Edit command to not open a file for editing a second time if that file is already being editing
- Redesigned Recent Folders menu to improve readability and display of folder hierarchy
- Added automatic retry when certain errors occur while deleting files
- Improved speed of deleting many files when Fetch is in the foreground
- Changed Command-O to be the keyboard shortcut for opening a remote folder, instead of opening a saved shortcut or mirror document
- Fixed a problem where Microsoft Office documents in the new Office XML Formats (such as .docx, .xlsx) were decoded into folders when downloaded
- Fixed a problem where the Back button would sometimes unexpectedly take you to the root folder instead of the previous folder

- Fixed incompatibilities between Fetch's Edit command and old versions of BBEdit on Mac OS X 10.5 and later
- Fixed rare VAT calculation errors in the purchase window
- Changed Automatic upload format to use Zip Archive instead of Stuffit X Archive for preserving resource forks
- Disabled the "If a file already exists: Pick a different name" option in the Put dialog for FTP uploads
- Added information to transcript about which upload format or download mode was used to transfer a file
- Improved compatibility with servers that reset the connections at the end of uploads
- Fixed Put, Open, and Save dialogs to have all the features normally found in file dialogs on Mac OS X 10.5 and later
- Digitally signed Fetch application with trusted certificate, to reduce unnecessary firewall alerts
- Added "Quit Fetch after upload" option to droplet shortcuts
- Added option to create droplet shortcuts that upload using a specific format
- Added support for saving a shortcut in the Fetch Shortcuts window as a droplet shortcut
- Added Open Folder item to Go menu
- Added Command-T as keyboard shortcut for New Shortcut command
- Improved compatibility of Unicode (UTF-8) preferred encoding
- Added capability to display file sizes in terabytes (TB), petabytes (PB) and exabytes (EB)
- Changed Command-double-clicking on folders so that a new window opens only when a single folder is selected
- Changed list of Bonjour servers to exclude Macintosh that Fetch is running on
- Enabled Put command when a folder shortcut is selected
- Disallowed getting info on a shortcut
- Removed Get Quick Info command because the Get Info command no longer calculates folder sizes automatically, and thus is quick by default
- Disallowed pasting an item into itself when using copy and paste to transfer files
- Fixed a problem where selecting and then uploading nested items in Mac OS X's list view would cause them to be uploaded twice
- Fixed a problem where the info window and file list cache date sometimes did not honor the date and time format settings from System Preferences
- Fixed a problem displaying the size of extremely large files in DOS-style file lists
- Fixed crash when trying to list files with extremely long names
- Fixed a problem saving edited files with CSSEdit on Mac OS X 10.5 and later
- Fixed a problem that caused certain PageSpinner files to be stuffed on upload
- Fixed a problem that caused certain disk images (.dmg) to be stuffed on upload
- Fixed a problem that caused certain AIFF files created by Logic to be stuffed on upload
- Fixed a problem that caused certain AppleWorks/ClarisWorks files to be stuffed on upload
- Fixed a rare problem where the Option key would be ignored when choosing from the Shortcuts menu
- Fixed failure to launch on Mac OS X 10.6 Snow Leopard
- Fixed crash on launch when Desktop folder does not exist
- Fixed rare crash when uploading folders
- Fixed rare stalls when downloading small files from certain servers
- Fixed rare case where SFTP downloads would stall with certain servers
- Fixed very rare hang connecting to certain servers
- Fixed rare freeze when resizing transfer window with certain toolbar configurations
- Fixed rare stalls when connecting to certain FTP with GSSAPI servers that don't support data encryption
- Improved warning dialog when connecting to an FTP with TLS/SSL server that doesn't

support encryption

- Fixed transcript window so that it doesn't unexpectedly scroll to the bottom
- Improved recognition of URLs and paths entered in the Hostname field
- Fixed a problem where extra white space wasn't removed from end of hostname and username fields in the New Connection dialog
- Made sure SFTP uploads and downloads transfer items in alphabetical order
- Changed name of Open command in file list and shortcut list window contextual menus to Open Folder
- Added Paste command to file list and shortcut list window contextual menus
- Added Put command to shortcut list window contextual menu
- Removed Get Info for Other command
- Added support for converting AppleScript objects to Fetch Items objects in Automator
- Fixed a problem where the Fetch Shortcuts contextual menu would not work when Fetch was in the background
- Fixed a rare problem where shortcut list window contextual menu choice would operate on transfer window instead of shortcut list
- Fixed a problem where clicked shortcut would not be selected when clicking on the Fetch Shortcuts window in the background
- Updated transcript to reflect new labels for firewall status in Mac OS X 10.5.1 and later
- Fixed a problem where the file list would sometimes not refresh correctly after Duplicate or copy/paste upload to an SFTP server
- Fixed a problem where filenames beginning with a period could still appear in the file list after checking "Omit filenames beginning with a period" preference
- Fixed a problem where transfer window wouldn't accept drags from certain applications (including iTunes and Address Book)
- Improved some error messages, including ones for SSL errors
- Fixed rare crashes when uploading ClarisWorks files with Automatic upload format
- Fixed a problem detecting folders on certain VShell SFTP servers
- Enabled proxy password field for USER fireID@remoteHost proxy type
- Fixed disappearing hostname when Proxy hostname field contained leading spaces
- Fixed incorrect Proxy port setting when Proxy hostname field contained leading spaces
- Changed Fetch debug logging to use our own log file (~/.Library/Logs/Fetch.log) instead of system.log, and to include transcript as well as debugging information

### Fetch 5.3.1

- Fixed problems with paid software updates
- Fixed purchase problems on future versions of Mac OS X

### Fetch 5.3

The major new features in Fetch 5.3 are:

- Improved compatibility with Mac OS X 10.5 Leopard
- Improved transfer window appearance on Mac OS X 10.5
- Redesigned toolbar and button icons with updated look for Mac OS X 10.5
- Added preference to [turn off automatic decoding](#) for all file types
- Digitally signed Fetch application, to reduce unnecessary keychain alerts
- Added support for Mac OS X 10.5's [application-specific firewall](#)
- Added support for [uploading files using Copy and Paste](#)
- Added support for [copying files on servers using Copy and Paste](#)

Other fixes and improvements in Fetch 5.3 are:

- Changed default download location to be the Downloads folder on Mac OS X 10.5
- Improved WebView's **Get from Safari** button to list all open tabs when used with Safari 3
- Improved recognition of URLs and paths entered in the Hostname field
- Fixed "Cannot convert from CFNumber to CFBoolean" errors when MobileMe (.Mac) Preferences syncing is enabled on Mac OS X 10.5
- Specified that the Fetch Automator actions should be listed under the Internet category on Mac OS X 10.5
- Specified that the Fetch Cache file should not be backed up by Time Machine on Mac OS X 10.5
- Marked passwords saved in the keychain by Fetch as "Fetch passwords"
- Disabled **FTP with KClient** connections on Mac OS X 10.5, because KClient (Kerberos v4) is no longer supported on 10.5
- Fixed a problem that caused certain QuickTime movies to be stuffed on upload
- Fixed problems saving shortcut list documents to SMB volumes and Windows-based Apple File Protocol (AFP) servers
- Added large (512 x 512) application and document icons for better appearance in Finder's CoverFlow view
- Added alternate row shading to various windows for Mac OS X 10.5
- Added everyday language error messages for new Mac OS X 10.5 network error codes
- Updated transcript window to display information about new firewall settings in Mac OS X 10.5
- Fixed problems saving changes to Automator actions on Mac OS X 10.5
- Fixed error when adding Generate Fetch Item Report action to workflows on Mac OS X 10.5
- Fixed error when dismissing Delete Fetch Items action confirmation dialog on Mac OS X 10.5
- Fixed Set Fetch Item Permissions Automator action to allow you to set permissions to "000"
- Fixed incorrect icons of downloaded files in Downloads stack on Mac OS X 10.5
- Improved compatibility of FTPS uploads with CrushFTP servers
- Changed icon displayed for aliases (symlinks) on servers to a combination folder/file icon
- Fixed a problem where the file list contextual menu did nothing when Fetch was in the background
- Fixed crash when trying to update a corrupted widget
- Fixed problems parsing pasted URLs containing scheme names in all capitals
- Fixed rare assertion errors when opening Preferences window
- Improved behavior of URLs for VM servers
- Improved reliability of software update checks when the current version is up-to-date
- Updated "Rebuild Launch Services database" script to work on Mac OS X 10.5

### Fetch 5.2.1

- Fixed slow transfers when Fetch is in the foreground on some Macintosh models
- Improved speed of Shortcuts and Go menus when they contain many items
- Fixed droplet shortcuts so they can be used with accounts limited by parental controls (managed accounts)
- Added information about a droplet shortcut's destination to the Finder Get Info window
- Fixed problems with WebView and usernames that contain underscores
- Fixed errors when creating a WebView entry and a "View Source" window is open in Safari
- Improved compatibility with user-assigned keyboard shortcuts
- Fixed -981 and -982 errors when opening shortcut list windows and transcript windows

with certain font setups

- Fixed a problem where Dashboard would sometimes be displayed without the user's permission when updating the Fetch widget
- Fixed a problem where some SFTP errors would not be reported
- Fixed empty file lists on some servers that don't support the LIST -al command
- Fixed a problem that caused some PageSpinner files to be stuffed on upload
- Fixed a problem where dates could be displayed as coming from the future on certain servers
- Fixed errors when checking for software updates through a proxy server
- Fixed incompatibility with Default Folder X and the path pop-up menu in the Put dialog
- Changed WebView to remove filenames from web addresses selected using the "Get from Safari" button
- Added Comcast FTP with TLS/SSL connections to the list of automatically created WebView entries
- Fixed a problem where Command-S was occasionally ignored the first time it was typed
- Fixed a problem where clicking on transfer window scroll bar would not make the window active
- Fixed problems entering multiple domains in the "Do not use proxy for" preference
- Fixed assertions when connecting after "Add filename extension when uploading" preferences had been changed
- Fixed rare -25308 error when connecting (caused by keychain problems)
- Fixed rare crashes when connecting to certain servers
- Fixed a crash caused by corrupt Fetch Cache file
- Changed Back command keyboard shortcut to Command-( in French translation
- Changed Command-double-clicking a folder to open the new transfer window in front of, not behind, the existing window
- Improved compatibility with CrushFTP Bonjour servers
- Improved some error messages
- Removed space from default name of new Stuffit archives to improve compatibility
- Disabled minimizing of Find window

## Fetch 5.2

The major new features in Fetch 5.2 are:

- Added [WebView](#) — easily view files in your web browser and copy web addresses while in Fetch
- Added support for [FTP with TLS/SSL \(FTPS\)](#) connections (requires Mac OS X 10.4 or later)
- Added [droplet shortcuts](#) — drag files to a droplet shortcut's icon to upload them to your server
- Overhauled FTP networking for improved compatibility and performance
- Moved [Refresh](#) button from an optional toolbar button to a permanent button on transfer window info bar
- Added [Get As](#) command that prompts you to choose a folder in which to save downloads

Other fixes and improvements in Fetch 5.2 are:

- Fixed stalls when downloading large files from certain servers
- Fixed crash when using Get Specified Fetch Items action in an Automator workflow saved as an application
- Made Fetch the default FTP with TLS/SSL application when it is the default FTP application
- Fixed connections using SOCKS4 on Intel-based Macintosh computers

- Improved support for passive mode connections to servers through NATs with port-mapping
- Fixed problems with line endings when pasting transcripts into Eudora
- Fixed problem where a shortcut would be deselected after it was edited
- Removed need to activate a transfer window before dragging files from it
- Fixed a problem saving files with long file names when editing with TextMate
- Added keyboard shortcut (Command-Option-Down arrow) to open New Connection dialog filled out with information from the selected shortcut
- Fixed incorrect display of some filenames when the names contained dates
- Fixed problems printing some graphics on Intel-based Macintosh computers
- Changed text windows to use preferred encoding preference when displaying text
- Changed file list to automatically select new folder after creating it
- Changed default retry between connections delay to 5 seconds
- Improved compatibility with DiskXtender/UniTree servers
- Improved compatibility with Tops-20 servers
- Improved compatibility with Windows CE servers
- Improved compatibility with Stuffit 11
- Improved handling of usernames, shortcuts, and URLs that contain non-ASCII characters
- Removed multi-part BinHex decoding support, fixing potentially incorrect error messages when downloading corrupt BinHex files
- Fixed editing of shortcut paths in shortcut list windows
- Fixed a problem that prevented the Window menu from listing the Mirror window when it was opened via a mirror document
- Fixed keyboard shortcuts for rotating through shortcuts in non-English languages
- Fixed error when clearing "new name" field in Rename Fetch Items Automator action
- Improved error handling in Set Fetch Item Permissions Automator action
- Improved feedback in Automator actions when there are no recent connections or shortcuts to choose
- Fixed confirmation dialog of Delete Fetch Items action so that Escape key is equivalent to Cancel button
- Improved handling of URLs for remote folders, fixing some unnecessary file lists redraws
- Fixed a problem that could cause duplicate entries in the file list when "Save downloaded files to" was set to "Ask every time"
- Fixed Edit Other to not save files in incorrect folder in certain circumstances
- Fixed a problem where the Edit New File dialog was sometimes incorrectly displayed instead of the Edit Other dialog
- Fixed potential crash when sending AppleScript commands to multiple transfer windows simultaneously
- Prevented certain rare files from being incorrectly decoded as MacBinary
- Fixed a problem with AppleScript recording of "default text font size"
- Enabled "hide extension" checkbox when saving transcript and other text windows
- Fixed Path pop-up menu and transfer window title bar so they display Home icon again
- Fixed problems mirroring between Kerberos servers and non-Kerberos servers

### **Fetch 5.1.1**

- Added support for SOCKS 5 password authentication
- Fixed a bug that prevented Fetch from opening on case-sensitive file systems
- Fixed a bug that prevented Fetch from opening on Mac OS X 10.3.9 when the BSD subsystem wasn't installed
- Fixed problems importing shortcut files that include broken shortcuts
- Fixed recent connections so they won't remember passwords that are not stored in the keychain

Fixed widget crash on some Intel-based Macintosh computers (including Mac Pros and iMac Core 2 Duos)

- Fixed crash in Get Specified Fetch Items action on some Intel-based Macintosh computers (including Mac Pros and iMac Core 2 Duos)
- Some Mac Pros shipped with Mac OS X 10.4.7 Build 8K1079, which includes a bug that causes Fetch 5.1 to crash when you open a connection; affected Mac Pros need to be upgraded to Mac OS X 10.4.8 in order to run Fetch 5.1.1 or later
- Fixed inaccurate display of size of newly uploaded files when using SFTP
- Fixed problems with the Upload Files action in Swedish
- Fixed layout problems with the Rename Fetch Items action in Spanish
- Fixed tab navigation in Set Fetch Item Permissions action
- Fixed widget problems on Mac OS X Leopard Developer Preview
- Fixed potential crash when editing hostnames in shortcut list windows
- Fixed potential crash when setting URLs of shortcuts via AppleScript
- Fixed crash when typing in a blank shortcut list window
- Fixed inaccurate font size display for new shortcut list windows
- Changed new shortcuts created with Make Shortcut to default to kind Folder, instead of kind Unknown
- Added skEdit as a supported text editor
- Fixed a problem that caused some GoLive files to be stuffed on upload
- Fixed drawing problems with some toolbar buttons when reducing transfer window width
- Fixed a problem where a Mirror document was saved with incorrect information after switching between local and remote tabs
- Fixed a problem where the incorrect folder could be mirrored after switching between local and remote tabs
- Fixed problems duplicating shortcut list windows with AppleScript

## Fetch 5.1

The major new features in Fetch 5.1 are:

- Added support for Intel-based Macintosh computers — Fetch is now a Universal binary
- Added [Fetch Dashboard widget](#), for easy uploading and checking on the progress of your transfer (the widget is installed automatically when you start Fetch)
- Added [Automator actions](#) for Fetch, including Upload Files, Download Files, Set Fetch Item Permissions, Rename Fetch Items, Move Fetch Items, and more
- Made the [New Connection dialog](#) modeless — you can now use other Fetch windows while the New Connection dialog is open
- Included StuffIt Engine with Fetch application — separate installation or updating of StuffIt is no longer required to use StuffIt support in Fetch
- Changed [Zip Archive format](#) to preserve Macintosh information (Fetch creates .zip files compatible with the Finder's Compress/Create Archive command)
- Changed shortcuts to always store passwords in the keychain (passwords stored in shortcuts will be moved to the keychain when you first open Fetch 5.1)

Other fixes and improvements in Fetch 5.1 are:

- Added "Clear menu" item to the Recent Connections menu in the New Connection dialog
- Changed "Do not upload dot files in folders" preference to apply to both files and folders whose names begin with periods
- Added Shortcuts and Recent Connections submenus to Fetch Dock icon's menu
- Fixed potential crashes when errors occur while Fetch is in the background
- Fixed high CPU usage when errors occur while Fetch is in the background

- Fixed problems launching Fetch due to bad fonts
- Improved compatibility with ProFTPD servers
- Improved compatibility with SnapOS servers
- Added AppleScript support for reading transfer time remaining, transfer rate, and transfer percent done
- Expanded AppleScript support for URLs and remote item properties
- Fixed a problem with AppleScript access when Fetch is hidden
- Fixed potential -30006 error when using "interarchy edit" AppleScript command
- Added .cfm (ColdFusion) files to the list of recognized text files
- Fixed a problem that caused some PageSpinner files to be stuffed on upload
- Fixed a problem that caused some files containing Suitcase resources to be stuffed on upload
- Fixed a problem where some shortcut files for home folders opened to the root folder instead
- Fixed a crash caused by corrupt Fetch Cache file
- Fixed a problem importing Interarchy 8 bookmarks
- Fixed a problem where newly moved items were sometimes not shown in the file list
- Fixed unnecessary file list updates
- Changed upload behavior to refresh file lists before uploads begin, instead of afterwards
- Fixed a problem with the Set Upload Permissions command
- Fixed incorrect status message when a server-to-server transfer is interrupted
- Fixed mirror transfer failures caused by folders with quotes in their name
- Fixed missing characters when displaying long messages from servers
- Added support for specifying KClient and GSSAPI connections in FTP URLs
- Improved handling of URLs with empty passwords
- Fixed potential error when doing server-to-server mirror transfers from AppleScript
- Fixed problems with Copy URL and Enable/Disable Encryption keyboard shortcuts
- Improved tab order in Password prompt dialog
- Changed "Welcome to Fetch" dialog to display price of Fetch
- Changed file list to automatically select child folder after moving to parent
- Stopped scroll wheel from scrolling the active window when pointer is outside the window
- Made heart icon in New Connection dialog darker

## Fetch 5.0.5

- Fixed incorrect icons of downloaded files on Mac OS X 10.4.3
- Fixed incorrect display of file size for downloaded files on Mac OS X 10.4.3
- Fixed incorrect icons of Microsoft Word (.doc) and Excel (.xls) files after downloading
- Stopped adding resource forks to downloaded files unnecessarily, which was causing some downloaded files to be larger than they were on the server
- Improved compatibility with Windows VanDyke VShell SFTP servers
- Fixed quit on launch caused by some Bonjour (Rendezvous) setups
- Fixed a problem that caused some ClarisWorks files to be stuffed on upload
- Fixed a problem where cleared Recent Connections did not stay cleared unless you made another connection
- Improved detection of helper applications
- Fixed errors caused by removing a helper application while Fetch is running
- Improved error message when a helper application cannot be found
- Improved wording of French error messages
- Made it easier to troubleshoot firewall conflicts
- Fixed SFTP connections under Rosetta
- Improved Unicode compatibility in shortcuts

- Made disk image background display even when default is column view
- Fixed a problem where the name of the read me file was sometimes displayed as "ReadMe"
- Stopped displaying "delete stray items at destination" warning when opening mirror documents
- Removed automatic decoding of Unix compress (.Z) files
- Fixed "Add filename extension when uploading text files" and "Add filename extension when uploading other files" preferences
- Fixed error -666 reading corrupt Internet Config settings
- Fixed a problem where files with unknown extensions would try to open in Fetch after downloading
- Fixed a problem parsing certain file names that look like dates
- Fixed a problem that sometimes prevented URLs copied from transfer windows from being pasted into shortcuts windows
- Fixed error -30200 when using GSSAPI to connect to some servers
- Improved compatibility with ProFTPD's mod\_gss security
- Improved compatibility with OnMyCommand

#### Fetch 5.0.4

- Fixed problem with purchasing licenses
- Fixed automatic decoding of BinHex files
- Made adding filename extensions when uploading BinHex and MacBinary files the default again

#### Fetch 5.0.3

The major enhancements in Fetch 5.0.3 are:

- Added Dutch, French, German, Italian, Spanish, and Swedish localizations
- Made Fetch Help available as a [downloadable PDF](#)
- Added support for many more language encodings for displaying file lists, including Arabic and Simplified Chinese
- Fixed dog cursor so that it runs again during transfers
- Improved compatibility with VMS FTP servers
- Fixed delays going to sleep when Fetch was running

Other fixes and improvements in Fetch 5.0.3 are:

- Changed file list to not display VMS version numbers
- Increased precision of file size and amount transferred displays for files larger than 1 GB
- Fixed a problem where long file names could be scrambled on upload to Rumpus servers
- Fixed a problem with resuming downloads after a crash
- Fixed the "Use case-sensitive sorting" preference
- Fixed error -108 when choosing a Classic application as a helper application
- Fixed a problem that caused some EPS files to be stuffed on upload
- Fixed Put dialog so that it is no longer dismissed when the "Put As" field is clicked on Mac OS X 10.2
- Fixed folder navigation on UNIX servers that do not have a / root folder
- Fixed a problem that caused the Fetch Cache file to grow too large
- Fixed problems reading corrupt Fetch Cache files
- Fixed problems reading corrupt Internet Config settings
- Fixed problems launching Fetch on non-English systems after changing default text window font

- Fixed problems using local paths that contain non-ASCII characters in mirror documents
- Fixed problems opening shortcut and mirror documents after changing encodings
- Fixed a problem with disappearing shortcuts after changing encodings
- Fixed problems reading recent connections after changing encodings
- Fixed an error when quitting from AppleScript
- Fixed inconsistent display of port number in Proxy preferences pane
- Fixed a problem where the Mirror window would not remember its position after opening a mirror document
- Eliminated unnecessary password prompting when mirroring
- Fixed a problem where BBEdit was sometimes not selected as the default text editor
- Improved handling of StuffIt decoding errors
- Prevented pasting of carriage returns into filenames
- Fixed invalid hostname errors when importing certain Transmit favorites
- Fixed error -32768 on long downloads
- Fixed rare error -4960 messages when opening Fetch Shortcuts
- Fixed rare "port number is invalid" errors on launch
- Fixed rare "username is invalid" error on launch
- Fixed slow typing in Mirror connection dialog on Mac OS X 10.2
- Improved compatibility with OS/2 FTP servers
- Improved compatibility with Chyron character generators

### Fetch 5.0.2

- Fixed delays when finding helper applications
- Fixed a conflict with Ittec that caused the New Connection dialog to be dismissed at startup
- Fixed a rare problem where SFTP passwords were displayed while being typed
- Fixed a problem with blank file lists from WebSTAR servers
- Added support for wide-area Bonjour
- Added script for fixing problems with locating helper applications

### Fetch 5.0.1

- Improved reliability of uploads on Mac OS X 10.4
- Improved behavior of "Open files like this with" on Mac OS X 10.4
- Improved detection of helper applications
- Fixed problems when upgrading from certain Fetch 4 configurations
- Fixed crash caused by certain StuffIt configurations
- Fixed a problem launching Fetch when Courier or Monaco font is disabled
- Fixed a problem with incorrect dates on Windows servers
- Fixed a problem mirroring between two accounts on the same server
- Improved compatibility with MultiNet VMS SFTP servers
- Made behavior of Mirror window direction buttons less confusing
- Added .css to the list of recognized text files

### Fetch 5.0

The major new features in Fetch 5 are:

- Added support for transferring files using [SFTP](#) (SSH File Transfer Protocol)
- Redesigned the file transfer interface to include:
  - New [toolbar](#) for easy access to common Fetch functions
  - Back and Recent folders buttons for easier navigation
  - [File list](#) that works more like the Finder list view

Compact [status pane](#) at the bottom of transfer window

- Improved the Fetch user experience on Mac OS X
- Reworded error messages to use everyday language
- Reorganized [menus](#) for clarity
- Added context-sensitive help, including help tags and extensive Apple Help book
- Added Automatic Passive Mode — automatic detection of proper connection mode for more reliable transfers through firewalls and NAT devices
- Added support for [uploading files as Stuffit X archives](#), Zip archives, Gzip, or Gzipped Tar archives
- Added support for [automatically decoding](#) Stuffit X, Zip, Gzip, Gzipped Tar, and many more formats when downloading
- Added support for [resuming uploads](#) of binary file transfers
- Added [Bonjour \(Rendezvous\)](#) support — Fetch will automatically detect and list FTP or SFTP servers available via Bonjour (formerly called Rendezvous)
- Made file transfers faster
- Added a file transfer progress indicator to the Dock icon
- Added a recent connections pop-up menu to the [New Connection dialog](#)
- Added [automatic checks for updates](#) to Fetch, with easy update downloading
- Added an easier, more powerful interface for [setting file types, formats, and transfer options](#)
- Improved support for [non-ASCII and Unicode filenames](#)
- Improved compatibility with non-English servers

Other fixes and improvements in Fetch 5 are:

- Enhanced the [View](#) command to automatically determine if a file is text or media
- Enhanced the [Edit](#) command to automatically determine if a file is text or an image
- Improved the [Edit New File](#) command so that it will not replace existing files
- Improved PDF viewing and printing
- Improved reliability when working with long file names and paths
- Improved Windows server compatibility
- Added support for in-place [renaming of files](#) and editing shortcut information
- Added support for importing [Transmit favorites](#) and [Interarchy bookmarks](#)
- Added support for automatically importing Transmit favorites and Interarchy bookmarks if no Fetch Shortcuts exist
- Enhanced the [Recent Connections](#) menu to remember and display last folder visited
- Added support for Command-double-click to open a folder in a new window
- Added support for Command-clicking on transfer window titles to display a path pop-up menu
- Added support for Command-clicking on info window titles to display a path pop-up menu
- Added support for creating shortcuts by dragging transfer window or info window proxy icons
- Streamlined the interface for [putting](#) (uploading) files
- Changed upload order so that items upload alphabetically
- Improved performance when using drag-and-drop with a large number of items
- Added support for automatically scrolling file lists when dragging items
- Improved Energy Saver compatibility
- Added a bigger, smoother, colorful running dog cursor
- Improved drag-and drop-feedback — Fetch now shows standard drag-and-drop cursors
- Added support for setting UNIX [permissions](#) in numeric, ls, or symbolic format
- Improved [AppleScript support](#)

- Improved AppleScript recording
- Improved Kerberos and GSSAPI compatibility and reliability
- Improved VMS compatibility and support
- Improved VM/CMS compatibility
- Improved VSFTPD compatibility
- Improved compatibility with FileZilla servers
- Improved compatibility with WFTPD servers
- Improved compatibility with intrusion detection firewalls
- Improved reliability of long-duration transfers
- Made mirror transfers more robust
- Improved accuracy of transfer statistics
- Improved reliability of BinHex decoding
- Improved reliability of viewing QuickTime VR panoramas
- Reorganized [preferences](#)

The interface and behavior of Fetch 5.6 are similar to previous versions of Fetch 5.

The major changes in Fetch 5.6 are automatic resume of stalled or failed downloads, and improvements to the automatic resuming of uploads.

The major changes in Fetch 5.5 were the ability to edit any kind of file using any application and the changes will automatically be saved back to the server, automatic resume of stalled or failed uploads, and Quick Look support for previewing files on servers.

The major changes in Fetch 5.3 were improved compatibility and appearance on Mac OS X 10.5 Leopard, support for new Mac OS X 10.5 features and technologies, and a preference to turn off automatic decoding for all file types.

The major changes in Fetch 5.2 were the new WebView feature, support for FTP with TLS/SSL (FTPS), droplet shortcuts, and a rewritten FTP networking layer.

The major changes in Fetch 5.1 were that Fetch became Universal (built to work on both Intel-based and PowerPC-based Macintosh computers), and the addition of the Fetch widget, Automator actions, and built-in StuffIt support.

Here's a quick guide to the changes to existing features in Fetch 5.6. This is just an overview; for a detailed list of all the changes in Fetch 5.6, see the [Fetch 5.6 Release Notes](#). If you are upgrading from Fetch 4, see the [Fetch 5.6 for Fetch 4 Users](#) topic instead.

## Changes from 5.5 to 5.6

### Get and Download

Fetch now automatically tries to resume downloads when the transfer stalls or unexpectedly stops due to a network or server error. The manual [resume download](#) feature remains available, but you should rarely need it.

### Put and Uploading

Fetch's automatic resume upload feature now resumes uploads and avoids unnecessary retries in more circumstances.

### Droplet Shortcuts

Droplet shortcuts now queue uploads. If an upload to the server referred to by the droplet shortcut is already in progress when you drag items to the droplet shortcut, they will be uploaded after the current upload finishes, instead of opening a new transfer window.

### System Requirements

Fetch 5.6 now requires Mac OS X 10.4 or later. Mac OS X 10.3.9 is no longer supported.

## Changes from 5.3 to 5.5

### New Feature: Edit with Anything

The Fetch [Edit](#) command now lets you edit any kind of file using any application, and the changes will be saved back to the server automatically. This means the **Edit** command now works with any HTML editor (including Dreamweaver and GoLive), any image editor (including Photoshop), word processors (such as Microsoft Word), and any text editor (not just the limited set previously supported).

The default editor for a kind of file is usually the same as the application that would normally open that kind of file. If Fetch's default choice is not the application you would like to use for editing, hold down the Option key when choosing the **Edit** menu item or clicking the **Edit** button, and the command will become [Edit With](#). Fetch will ask you to choose a different application to use for editing and whether you want Fetch to remember this choice for the future or not.

You can also choose a different editor temporarily with the [Edit With submenu](#) in the **Remote** menu or the contextual menus.

The **Edit text files with** and **Edit image files with** preferences have been removed. Instead, you can [change the editor application](#) for a specific kind of file using the [Transfer Options](#) pane of the [info window](#).

### New Feature: Quick Look

On Mac OS X 10.5 and later, [Quick Look](#) has replaced the **View** command and toolbar button for previewing files without leaving Fetch. The **Quick Look** command can display many more kinds of files than the **View** command could — it can display text files, images, sounds, movies, PDFs, Microsoft Word and Excel files, and more. You can still view any file as a plain text file using [View as Text](#).

### Put and Uploading

Fetch now automatically tries to resume uploads when the transfer stalls or unexpectedly stops due to a network or server error. The manual [resume upload](#) feature remains available, but you should rarely need it.

When the [Automatic upload format](#) encounters a file that includes significant special Macintosh information that needs to be preserved (the file has a significant resource fork), Fetch will now upload that file using the Zip Archive format instead of Stuffit X Format.

The **If a file exists: Pick a different name** option in the [Put](#) dialog is no longer available for FTP uploads, because many FTP servers responded poorly to that option. It remains available for SFTP uploads.

### Get Info and Info Window

The [Get Info](#) command no longer automatically calculates size of folders, making it much quicker to open info windows, such as when you just need to set permissions. You can calculate the size of folders by clicking the new **Calculate** button in the [info window](#). The **Get Quick Info** command has been removed since it is no longer

necessary.

## Transfer Window

The display of information in the transfer window title bar has changed in two ways. First, the title bar shows the shortcut name instead of the server's hostname when there is a matching shortcut for that server. Second, the folder name is now listed first, followed by the shortcut name or hostname.

On laptops whose trackpads support gestures, in the transfer window you can now swipe left to go back to previously viewed folder, swipe up to go to the parent folder, and swipe down to open the selected folder.

The transfer window now remembers your last column sorting choice, so if you prefer that your file lists always start sorted by modification date, they will be.

You can now type Command-O to open the selected folder in the file list.

## Recent Connections

The Recent Connections menu now shows the shortcut name instead of the hostname when there's a matching shortcut.

## Droplet Shortcuts

There are two new options when you create a [droplet shortcut](#) — you can set the droplet shortcut to always upload files in a specific [upload format](#), and choose to have Fetch quit after an upload started by the droplet shortcut completes.

## Changes from 5.2 to 5.3

### New Features

There is now a preference to turn off automatic decoding for all file types. Choose **Fetch > Preferences**, click the **Download** tab, and check or uncheck the **Allow automatic decoding of downloaded files** checkbox. When the **Allow automatic decoding of downloaded files** checkbox is checked, you can continue to disable decoding for only certain kinds of files using the **Automatically decode files like this** checkbox in the [Transfer Options](#) section of the [info window](#).

You can now [upload files and folders using Copy and Paste](#), or [use Copy and Paste to copy files and folders](#) from one folder to another on the same server, or from one server to another.

### Mac OS X 10.5 Leopard Support

The transfer window and toolbar buttons have an updated look for Mac OS X 10.5.

The Fetch application is digital signed, which among other things, will

reduce or eliminate various keychain alerts.

Fetch supports the Mac OS X 10.5 [application-specific firewall](#).

## New Connection Dialog

You can now enter a path after the hostname in the Hostname field, e.g. ftp.fetchsoftworks.com/example, and be taken to the folder specified by that path.

On Mac OS X 10.5 and later, the **FTP with KClient** choice in the **Connect using** pop-up menu is disabled, because KClient (Kerberos v4) is no longer supported on Mac OS X 10.5.

## Preferences

The **Allow automatic decoding of downloaded files** checkbox has been added to the [Download Preferences](#) pane.

On Mac OS X 10.5 and later, the default download location as specified by the **Save downloaded files to** preference is the new Downloads folder. On Mac OS X 10.4, the default download location remains the Desktop.

## Automator

On Mac OS 10.5 and later, Automator lists the [Fetch actions](#) under the Internet category.

## Changes from 5.1 to 5.2

### New Features

[WebView](#) lets you easily view files in your web browser and copy web addresses while in Fetch. Once you've created a WebView entry that sets up a correspondence between the Fetch address of items on a file server and the web address of the webpage they are part of, you can easily view changes to your site in a web browser by clicking the **WebView** button in Fetch. You can also use the [Copy Web Address](#) command to copy a web address link (URL) of the items selected in Fetch. See the [WebView](#) help topic for more information.

Fetch now supports connecting using **FTP with TLS/SSL** (FTPS) — choose **FTP with TLS/SSL** from the **Connect using** pop-up menu. See the [FTP with TLS/SSL](#) help topic for more information. If Fetch is your default FTP helper, it will become your default FTP with TLS/SSL helper automatically.

You can use the new [Save Droplet Shortcut](#) command to create a [droplet shortcut](#) to which you can drag and drop files to in the Finder or Dock to upload those files to a specific server.

The FTP networking layer has been rewritten for improved compatibility and performance.

## New Connection Dialog

The **Connect using** pop-up menu now lists **FTP with TLS/SSL** (FTPS) as one of the choices. Shortcuts and recent connections also support **FTP with TLS/SSL**.

## Transfer Window and Toolbar

The [Refresh](#) button is no longer in the transfer window toolbar; it is now a smaller button (↻) located on right-hand side of the info bar just above the file list.

A [WebView](#) button has been added to the toolbar.

You may lose some toolbar customizations when upgrading from an earlier version of Fetch to Fetch 5.2, due to the changes in the toolbar buttons. We apologize for the inconvenience.

## Menus

The [Save Droplet Shortcut](#) command is available in the **File** menu when you select no files or a single folder in the file list.

If you hold down the Option key when choosing [Get](#), the command will change to [Get As](#), and Fetch will prompt you to choose a folder on your Macintosh to which it should download the selected files and folders. This lets you temporarily override the download folder specified by the **Save downloaded files to** preference in the [Download Preferences](#) pane.

The **Copy URL** command has become [Copy Fetch Address](#) and [Copy Web Address](#).

The **Copy Plain** and **Copy Plain URL** commands have been removed; all URLs are now copied without angle brackets.

## Preferences

The Preferences window has a new pane, [WebView](#), for viewing and modifying your WebView entries.

The **Contact server during long transfers** preference has been removed from the Obscure pane, because it is no longer necessary as a result of the rewritten FTP networking layer.

## Changes from 5 to 5.1

### New Connection Dialog

The New Connection dialog is now modeless — you can use other Fetch windows and most Fetch menu commands while the New Connection dialog is open.

The Recent Connections  pop-up menu now has a **Clear Menu** command at the bottom of the menu for removing all the entries from the menu.

### Shortcuts and Passwords

Fetch 5.1 and later store shortcuts in a different format than previous versions of Fetch, so while you can continue to use Fetch 5.1 and an earlier version of Fetch on the same Macintosh, you will not see changes you make to the Fetch 5.1 shortcuts in the previous versions of Fetch, and vice versa. The first time you open Fetch 5.1, it will automatically read and upgrade your shortcuts from your previous version of Fetch.

While previous versions of Fetch allowed you to save a shortcut password in the shortcut itself, in Fetch 5.1 all shortcut passwords are saved in your [keychain](#). When you upgrade to Fetch 5.1, your shortcut passwords will be automatically moved to the keychain if they are not stored there already.

The file where Fetch stores your shortcuts has changed. They are now stored in a file named "com.fetchsoftworks.Fetch.Shortcuts.plist" in the Preferences folder in the Library folder in your home folder. Fetch 5 stored shortcuts in a file named "Fetch Shortcuts" in that same location. (After you upgrade, you will have both files — the older file is saved both as a backup, and in case you want to continue to use Fetch 5.)

If you have created separate [shortcut list documents](#), they will be upgraded to the new format the first time you open them with Fetch 5.1. After they have been upgraded, you cannot use them with earlier versions of Fetch anymore. Fetch does not make a backup during the upgrade, so if you think you may want to continue to use them with an older version of Fetch, please make a backup of your shortcut list documents before opening them in Fetch 5.1.

## Stuffit and Archive Support

The Stuffit Engine is now built into the Fetch application. A separate installation or updating of Stuffit is no longer required to use [Stuffit and Archive support](#) in Fetch.

In Fetch 5.1 and later, the [Zip Archive format](#) now preserves Macintosh information. That is, like the Compress (Create Archive) command in the Finder, Fetch's creates .zip files that preserve both data and resource forks, long file names, and other Macintosh Finder information.

If a Windows user downloads a Zip archive created by Fetch, it may contain extra data that cannot be interpreted on Windows. Windows users can easily ignore this extra Macintosh-only information. There is no way to create a Zip archive without the extra Macintosh information when uploading in Fetch, although the extra information is usually only added when necessary.

Fetch's automatic decoding will now properly decode Zip archives that contain Macintosh information.

## Fetch Widget

New to Fetch 5.1 is the Fetch Dashboard widget. The Fetch widget lets

you easily upload files and folders to a specific server by dragging and dropping items on the widget. After the upload has started, the Fetch widget displays the progress of the transfer. When you first open Fetch 5.1, you will be asked if you want to install the Fetch widget. For more information, see the [Fetch widget](#) and [using the Fetch widget for uploading](#) help topics.

## **Automator**

Also new to Fetch 5.1 are a number of actions for Mac OS X's Automator, allowing you to create simple workflows that can automate a variety of Fetch tasks. The actions are: Delete Fetch Items, Download Fetch Items, Filter Fetch Items, Generate Fetch Item Report, Get Fetch Folder Contents, Get Selected Fetch Items, Get Specified Fetch Items, Move Fetch Items, Rename Fetch Items, Set Fetch Item Permissions, and Upload Files. See the [introduction to using Automator with Fetch](#), [Automator actions](#) and [more examples of using Automator with Fetch](#) help topics for more information.

## **AppleScript**

The AppleScript dictionary in Fetch 5.1 has a variety of additions and changes in order to support new features and make Fetch even more scriptable than before. See the [AppleScript changes](#) help topic for more information.



Many changes and improvements have been made to Fetch 5.6. If you're a regular Fetch 4 user, this guide should help you find your favorite features in Fetch 5.6. Also be sure to read the [summary of new features in Fetch 5.6](#) to learn about new features that have been added since Fetch 4.

If you're already using Fetch 5, you may want to see the [Fetch 5.6 for Fetch 5 Users](#) guide instead.

## New Connection Dialog

The New Connection dialog is now modeless — you can use other Fetch windows and most Fetch menu commands while the New Connection dialog is open.

The Shortcuts pop-up menu is now to the right of the Hostname field, and is indicated by a heart icon: . The **Create New Shortcut** and **Create Default Shortcut** items have been combined as the [Make Shortcut](#) item.

Also next to the hostname field is the new Recent Connections pop-up menu, indicated by a clock icon: . The Recent Connections pop-up lists the last 10 connections you have made.

The new **Connect using** menu lets you choose the type of connection, including the new **SFTP** and **FTP with TLS/SSL** options, which use encryption to protect your password and data. See the [connection types](#) help topic for more information.

The new **Enable encryption** checkbox specifies whether your data will be encrypted as it is transferred. It is always on for SFTP connections, not available for FTP connections, and optional for FTP with TLS/SSL or Kerberos connections.

See the [New Connection dialog](#) help topic for more information.

## Transfer window

The transfer window has been redesigned with a new layout, a new toolbar, and a file list that works more like the Mac OS Finder's list view.

The **Get** and **Put** buttons have moved to the toolbar. The **Mode** and **Format** pop-up menus have been removed from the standard transfer window layout, because Fetch's **Automatic** mode and format should do the right thing almost all the time. However, you can add **Mode** and **Format** buttons to the toolbar with the **View > Customize Toolbar** command. You can also choose a new download mode or upload format from the **Remote** menu.

Navigating remote folders is easier with the **Back** and **Recent** folders buttons in the toolbar. The toolbar also has buttons for several other commonly used Fetch commands. You can add or remove toolbar buttons using the **View > Customize Toolbar** command, or hide it altogether with the **View > Hide Toolbar** command. See the [transfer window toolbar](#) help topic for more information about the toolbar and the buttons in it.

The progress donut, transfer status, and connection status have been moved to the new, compact [status pane](#) beneath the file list. The **Stop** button is now an icon: .



The path pop-up menu that used to be directly above the file list is now the **Path** button in the toolbar. (You can also Command-click the transfer window's title.) The name of the current folder, and then either the name of the shortcut matching the current server or the server's hostname, is now displayed in the title bar.

The Parent proxy icon (the little folder in the top left of the Fetch 4 transfer window) has been removed. To create a URL to the current folder, you can drag the proxy icon from the Fetch 5 transfer window's title bar to another application. You can also add a **Parent** button to the Fetch 5 toolbar with the **View > Customize Toolbar** command. You can then drag files and folders from the file list to the **Parent** button to move them to the parent folder.

You can now rename files directly in the file list by clicking once to select an item, and then pressing the Return key to show the edit box. Type a new name or edit the existing name, and press Return again.

Pressing the Return or Enter key no longer downloads the selected file or opens the selected folder; instead it allows you to rename a file. To download a file or open a folder using the keyboard, type Command-Down arrow. You can also open folders by typing Command-O.

See the [transfer window](#) and [file list](#) help topics for more information about working with the Fetch 5 transfer window.

## Putting Files

The interface for uploading multiple files and folders has changed. In Fetch 5, there is a single command for putting files and folders, regardless of whether you are putting one file, one folder, or many files or folders. You do not have to add files and folders to a list to be uploaded as you did in Fetch 4. Instead, in the Put dialog just use the Shift and Command keys to select multiple files and folders. You can also simply drag and drop files and folders to a Fetch transfer window to upload them.

The options for upload format and what to do if files with the same name exist on the server are now in the same dialog that you use for selecting the files and folders, instead of a separate dialog. Fetch now supports resuming interrupted uploads of certain kinds of files; see the [resuming uploads](#) help topic.

Fetch 5 supports uploading files as [Zip archives](#), [Stuffit archives](#), and other archive formats.

For more information, see the [Put](#) command help topic.

[Droplet shortcuts](#), the [Fetch widget](#), and [uploading files using Copy and Paste](#) are three new ways to upload files using Fetch 5.

## Getting Files

The options for what Fetch should do with files after they've been downloaded, such as which application they should open with, or whether they should be automatically opened or automatically decoded, are now part of the [Transfer Options](#) pane of the [info window](#). The Fetch 5 Transfer Options replace the Suffix Mapping and Post-Processing commands and Recognize and Decode preferences

of Fetch 4.

To set these options in Fetch 5, select a file whose name ends with an extension for the kind of file that you want to change the behavior of — such as a file ending in .txt for text files — and then choose **Remote > Get Info**, and go to the Transfer Options pane of the info window.

New to Fetch 5 and later is support for automatic decoding of Stuffit files and other compression and archive formats. Automatic decoding can be disabled for all files in the [Download Preferences](#) pane, or just for certain kinds of files in the Transfer Options pane of the info window.

For more information, see the [changing the application a file opens with](#), [opening files automatically after downloading](#), [decoding files automatically](#), and [Transfer Options](#) help topics.

## Setting Permissions

The Set Permissions command of Fetch 4 has been replaced by the Ownership and Permissions pane of the info window in Fetch 5. To set the permissions of a remote file or folder, select it, choose **Remote > Get Info**, go to the Ownership and Permissions pane, set the desired permissions, and click the **Apply** button.

For more information, see the [viewing and setting permissions](#) help topic.

## Shortcuts and Passwords

Fetch 5.6 stores shortcuts in a different format than Fetch 4, so while you can continue to use Fetch 5.6 and Fetch 4 on the same Macintosh, you may not see changes you make to the Fetch 5.6 shortcuts in the previous versions of Fetch, and vice versa. The first time you open Fetch 5.6, it will automatically read and upgrade your shortcuts from your previous version of Fetch.

While some previous versions of Fetch allowed you to save a shortcut password in the shortcut itself, in Fetch 5.6, all shortcut passwords are saved in your [keychain](#). When you upgrade to Fetch 5.6, your shortcut passwords will be automatically moved to the keychain if they are not stored there already.

The file where Fetch stores your shortcuts has changed. They are now stored in a file named "com.fetchsoftworks.Fetch.Shortcuts.plist" in the Preferences folder in the Library folder in your home folder. Fetch 4 stored shortcuts in a file named "Fetch Shortcuts" in that same location. (After you upgrade, you will have both files — the older file is saved both as a backup, and in case you want to continue to use Fetch 4.)

If you have created separate [shortcut list documents](#) in Fetch 4, they will be upgraded to the new format the first time you open them with Fetch 5.6. After they have been upgraded, you cannot use them with earlier versions of Fetch anymore. Fetch does not make a backup during the upgrade, so if you think you may want to continue to use them with an older version of Fetch, please make a backup of your shortcut list documents before opening them in Fetch 5.6.

## Menus

Fetch's menus have been extensively reorganized.

The **Remote** menu has mostly the same commands, but they have been renamed and reorganized.

The Fetch 4 **Directories** menu has been replaced by the new **Go** menu, although several commands have been moved to other menus.

Most of the commands from the Fetch 4 **Customize** menu have been superseded by new features; the shortcut commands have moved to the new **Shortcuts** menu.

The **View Text File** and **View Media File** commands have been replaced by the **Quick Look** command (on Mac OS X 10.5 and later) or combined into a single **View** command (on Mac OS X 10.4). The **Quick Look** and **View** commands automatically determines what kind of files are selected, and opens the correct type of window. On Mac OS X 10.5 and later, you can use the **View as Text** command to view a file as plain text, regardless of its contents or extension. On Mac OS X 10.4, hold down the Option key when choosing **View** to override the automatic behavior and choose whether to view a files as text or media.

The **Edit File with BBEdit** and **Edit File with GraphicConverter** commands have similarly been combined into a single **Edit** command, and Fetch can now edit any kind of file using any application on your Macintosh, instead of being limited to text and image files and only certain applications. If necessary, hold down the Option key when choosing **Edit** to select a different application for editing.

Here is a list of all menu commands that have been moved, renamed, or removed. If a menu command is not listed, it retained the same name and location. Select a command name for more information about it.

Old Command Location and Name	New Command Location and Name
<b>File &gt; New Shortcut List</b>	<b>Shortcuts &gt; <a href="#">New Shortcut List</a></b>
<b>File &gt; Open Shortcut</b>	Listed at bottom of <b>Shortcuts</b> menu
<b>Edit &gt; Set Font and Size</b>	<b>View &gt; <a href="#">Set Font and Size</a></b>
<b>Remote &gt; View Text File</b>	<b>Remote &gt; <a href="#">Quick Look</a></b> or <b>Remote &gt; <a href="#">View as Text</a></b> (or <b>Remote &gt; <a href="#">View</a></b> on Mac OS X 10.4)
<b>Remote &gt; View Media File</b>	<b>Remote &gt; <a href="#">Quick Look</a></b> (or <b>Remote &gt; <a href="#">View</a></b> on Mac OS X 10.4)
<b>Remote &gt; View File List</b>	<b>Remote &gt; <a href="#">View Verbose File List</a></b>
<b>Remote &gt; Edit File with BBEdit</b>	<b>Remote &gt; <a href="#">Edit</a></b>
<b>Remote &gt; Edit File with GraphicConverter</b>	<b>Remote &gt; <a href="#">Edit</a></b>
<b>Remote &gt; Get Directories and Files</b>	<b>Remote &gt; <a href="#">Get</a></b>
<b>Remote &gt; Put Folders and Files</b>	<b>Remote &gt; <a href="#">Put</a></b>
<b>Remote &gt; Mirror Folder</b>	<b>Remote &gt; <a href="#">Mirror</a></b>
<b>Remote &gt; Rename Directory or File</b>	Edit filenames directly in the file list, or use <b>Remote &gt; <a href="#">Get Info</a></b>
<b>Remote &gt; Delete Directory or File</b>	<b>Remote &gt; <a href="#">Delete Item</a></b>
<b>Remote &gt; Search</b>	Use <b>Remote &gt; <a href="#">Send FTP Commands</a></b>
<b>Remote &gt; Set Permissions</b>	Use <b>Remote &gt; <a href="#">Get Info</a></b>
<b>Remote &gt; Set Upload Permissions</b>	Use <b>Remote &gt; <a href="#">Send FTP Commands</a></b>

Directories > Change Directory	Go > <a href="#">Go to Folder</a>
Directories > Create New Directory	Remote > <a href="#">New Folder</a>
Directories > Refresh File List	View > <a href="#">Refresh</a>
Directories > Clear Directories Menu	Go > <a href="#">Clear Menu</a>
Directories > list of recently visited folders	Listed at the bottom of the <b>Go</b> menu
Customize > Preferences	Fetch > <a href="#">Preferences</a>
Customize > Internet System Preferences	No equivalent; go to the <b>System Preferences</b> and choose <b>Network</b>
Customize > Suffix Mappings	Select a remote file with the desired extension, choose <b>Remote</b> > <a href="#">Get Info</a> , and use the <a href="#">Transfer Options</a>
Customize > Post-Processing	Select a remote file with the desired extension, choose <b>Remote</b> > <a href="#">Get Info</a> , and use the <a href="#">Transfer Options</a>
Customize > Fetch as FTP Helper	Use the <b>Default FTP application</b> and <b>Default SFTP application</b> preferences in the <a href="#">General Preferences</a> pane
Customize > New Shortcut	Shortcuts > <a href="#">New Shortcut</a>
Customize > Edit Shortcut	Shortcuts > <a href="#">Edit Shortcut</a>
Customize > Remove Shortcut	Edit > <a href="#">Delete</a>
Window > Fetch Help	Help > Fetch Help

Most menu keyboard shortcuts are the same, except for **Put**, which has changed from Command-T to Command-U; and **Change Directory** (Command-D) has been replaced by **Go To Folder** (Command-Shift-G).

## Mirror window

In the Mirror window, the pop-up menu at the top of each pane has been replaced with tabs, making it clearer which pane refers to a local folder and which refers to a remote folder. The right hand pane is now remote-only; if you have a saved mirror document that assumed a local pane on the right, it will be switched around automatically when it's opened.

To choose a local folder or remote folder for mirroring in Fetch 5, click the **Choose** button instead of choosing from the pop-up menu as you did in Fetch 4. To choose a currently open connection as a remote destination in Fetch 5, click the small pop-up menu to the left of the Host label in a Remote pane, again, instead of choosing from the big pop-up menu as you did in Fetch 4.

See the [Mirror](#) window help topic for more information.

## Fetch Shortcuts

In Fetch 5, the Host and Userid columns have been renamed to Hostname and Username, respectively. The Size and Date columns have been removed, as Fetch 5 no longer tracks that information for shortcuts. The security indicator is now simpler, showing either a locked padlock (secure encrypted connection), an unlocked padlock (securely authenticated, but unencrypted connection), or nothing (not secure). To find out which connection type a secure shortcut uses, you must

edit the shortcut.

For more information, see the [Fetch Shortcuts](#) and [shortcut list window](#) help topics.

## **AppleScript and Automator**

The AppleScript dictionary in Fetch 5 has changed to accommodate new features and changes in the user interface. See the [AppleScript changes](#) help topic for more information.

Fetch 5 features a variety of new [Automator actions](#) that you can use to [create simple workflows](#) that can automate a variety of Fetch tasks.

The AppleScript dictionary in Fetch 5.6 has been updated to accommodate new features and changes in the user interface. Where possible, Fetch 5.6 supports both the old and new terminology so that old scripts will still work with Fetch 5.6, but your scripts may need updating. To assist you, below is a summary of the changes.

See the [Scripting](#) help topic for more general information about using AppleScript with Fetch, and see the AppleScript dictionary for more details on the syntax for these terms.

[Changes in Fetch 5.6](#)  
[Changes in Fetch 5.5.2](#)  
[Changes in Fetch 5.5](#)  
[Changes in Fetch 5.3](#)  
[Changes in Fetch 5.2](#)  
[Changes in Fetch 5.1](#)  
[Changes in Fetch 5](#)

## Changes in Fetch 5.6

### Changes to parameters

[immediate return](#) (of the [put into](#) command) – If an upload to the same server is already in progress, now queues the upload instead of opening a new window, in addition to returning immediately. However, uploads to a new server will still open new windows. (See [changes in 5.1](#) for more notes about the [immediate return](#) parameter.)

## Changes in Fetch 5.5.2

### Changes to commands

[view text](#) – Available on all versions of Mac OS X again (including Mac OS X 10.5 and later), because the **View as Text** command was added in Fetch 5.5.2.

## Changes in Fetch 5.5

### New commands

[edit new text file](#) – Create and edit a remote text file, either using the editor specified for the filename extension of the new file (as specified in the [Transfer Options](#) section of the [info window](#)), or using the application specified by the [using](#) parameter.

### New parameters

[using](#) (of [edit](#) command) – Specify the application to edit a file with. Fetch now supports editing of any kind of file with any application. When the using parameter is omitted, the edit command will open the file with the editor specified for the filename extension of the file (as specified in the [Transfer Options](#) section of the [info window](#)). The [using](#) parameter lets you temporarily use a different application for editing a file. The application can be specified either by the application name or by a path to the application.

## Changes to commands

[view](#), [view media](#) – Return errors on Mac OS X 10.5 and later, because the general **View** command is no longer supported for media on Mac OS X 10.5 and later. Note that the [view text](#) command remains available, because **View as Text** is available (in Fetch 5.5.2 and later).

## Removed commands

[edit text file](#) – Fetch now supports editing of any kind of file with any application, so this command has been superseded by the [using](#) parameter of the [edit](#) command (to edit something as a text file, tell Fetch to [edit](#) it [using](#) a text editing program).

[edit image file](#) – Fetch now supports editing of any kind of file with any application, so this command has been superseded by the [using](#) parameter of the [edit](#) command (to edit something as an image file, tell Fetch to [edit](#) it [using](#) an image editing program).

## Removed properties

[external text editor](#) (of class [Application](#)) – Since Fetch now supports editing of any kind of file with any application, it is no longer limited to specifying a single editor application for all text files. A different application can be chosen for each kind of file.

[external image editor](#) (of class [Application](#)) – Since Fetch now supports editing of any kind of file with any application, it is no longer limited to specifying a single editor application for all image files. A different application can be chosen for each kind of file.

## Changes in Fetch 5.3

### New properties

[allow automatic decoding](#) (of class [Application](#)) – When true, Fetch will try to decode or expand files automatically on download, as specified by that kind of file's **Automatically decode files like this checkbox** in the [info window](#). When false, downloaded files will never be automatically decoded or expanded by Fetch.

## Changes in Fetch 5.2

### New constants

[FTPS](#) – A new constant for the [protocol](#) property of many classes. Specifies the FTP with TLS/SSL connection type. See the [FTP with TLS/SSL](#) help topic for more information.

[TLS](#) – A new constant for the [authentication](#) property of many classes and commands. Specifies the FTP with TLS/SSL connection type. See the [FTP with TLS/SSL](#) help topic for more information.

### New commands

[drop on](#) – Handle the dropping of files on a [droplet shortcut](#).

### New properties

[web url](#) (of class [remote item](#), class [remote file](#), class [remote folder](#), class [url](#)) – The web address (URL) that corresponds to the referenced item, if [WebView](#) has been set up for the server where the item resides. If [WebView](#) has not been set up for the server, you will get an error when trying to access this property.

## Changes in Fetch 5.1

### New classes

[recent connection](#) – A class for referring to recent connections as listed in the **Recent Connections** menu; similar to the [shortcut](#) class.

[transfer](#) – A class for referring to a transfer, either an upload or a download of one or more items. Separate from the [transfer window](#) class, which refers to the window a transfer is taking place in. While the [transfer window](#) and [transfer](#) classes have some of the same properties, they also have properties unique to each class.

### New commands

[stop](#) – Stop the transfer referred to by a [transfer window](#) or [transfer](#).

### New parameters

[immediate return](#) (of the [open](#) command and [put into](#) command) – Have the event immediately return a reference to an object of class [transfer](#) instead of waiting for a reply.

[to file](#) (of [download](#) command) – *file* specifies the folder where the downloaded items will be saved.

### New properties

The [remote item](#), [remote file](#), and [remote folder](#) classes gained many new properties. They are not all listed here; consult the AppleScript dictionary to see the complete list of properties now supported by these classes.

[connected](#) (of class [transfer window](#)) – Indicates whether a transfer window is currently connected.

[home folder](#) (of class [transfer window](#)) – The home folder of the account on the server a transfer window is connected to.

[id](#) (of class [shortcut](#), class [recent connection](#), and class [transfer](#)) – A unique ID that can be used to refer to the shortcut, recent connection, or transfer.

[name extension](#) (of class [url](#), class [remote item](#), class [remote file](#), and class [remote folder](#)) – The file name extension of the item referenced.

[properties](#) (of class [url](#), class [shortcut](#), class [remote item](#), class [remote file](#), and class [remote folder](#)) – A record containing all the properties of the item.

[port](#) (of class [transfer window](#), class [shortcut](#), class [remote item](#), class [remote file](#), and class [remote folder](#)) – The port number of: the server a transfer window is connected to; the server referenced by a shortcut;

or the server where an item resides.

[protocol](#) (of class [transfer window](#), class [shortcut](#), class [remote item](#), class [remote file](#), and class [remote folder](#)) – The protocol being used for a transfer window's connection; the protocol specified by a shortcut; or the protocol being used to access an item. In 5.1, the possible values for [protocol](#) are [FTP](#) or [SFTP](#).

[transfer percent done](#) (of class [transfer window](#) and class [transfer](#)) – Percentage of the current item that has been transferred so far.

[transfer rate](#) (of class [transfer window](#) and class [transfer](#)) – The transfer rate for the transfer of the current item, in bytes per second.

[transfer time remaining](#) (of class [transfer window](#) and class [transfer](#)) – The estimated amount of time, in seconds, that remains until the transfer of the current item completes.

### New results returned

The [download](#) command now returns a list of download files.

The [put into](#) command now returns a list of records, containing a list of the properties of the uploaded files.

### New Automator actions

Fetch now includes a variety of Automator actions that may be of interest to AppleScript users. For more information, see the [introduction to using Automator with Fetch](#) and [Automator actions](#) help topics.

### Renamed commands

Fetch 5	Fetch 5.1
<a href="#">list</a> (in Interarchy Suite)	<a href="#">interarchy list</a>

### Changes to properties

[elapsed transfer time](#) (of class [transfer window](#) and class [transfer](#)) – Now returns the time in seconds, instead of 60ths of a second.

[name](#) (of class [remote item](#), class [remote file](#), class [remote folder](#)) – In these classes, the [name](#) property is now writable; previously it was read-only. In most other classes, the [name](#) property remains read-only.

### Changes to results returned

The [get information about command](#) now returns a list of records. There is a record for each file or folder referred to by command, and each record contains many properties about that item. It used to return just a simple list containing the folder count, file count, byte count of the items you were getting information about.

## Changes in Fetch 5

### New classes

[media window](#) – A class for referring to [media windows](#) that display

images, movies, and sounds. This class has no properties beyond those inherited from the [window](#) class. (Fetch media windows are used only on Mac OS X 10.4.)

### New commands

[edit](#) – Open a remote file in a helper application for editing. See the [Edit](#) command help topic for more information.

[view](#) – Open a remote file (text or media) for viewing in Fetch. See the [View](#) command help topic for more information. (Only available on Mac OS X 10.4.)

### New properties

[ask about unknown hosts](#) (of class [Application](#)) – Ask whether Fetch should continue to connect (via SSH) to a host whose host key Fetch has not seen before. See **SFTP: Ask before accepting unknown host keys** in the [Security Preferences](#) help topic for more information.

[automatically check for updates](#) (of class [Application](#)) – When true, Fetch will periodically check to see if any updates for Fetch are available. See **Automatically check for updates** in the [Miscellaneous Preferences](#) help topic for more information.

[permissions](#) (of classes [remote file](#), [remote folder](#), [remote item](#)) – The octal UNIX-style permission code for a file or folder (such as 755).

[preferred encoding](#) (of class [Application](#)) – The preferred encoding and character set for displaying non-ASCII file and folder names. See the [Character Encoding & Translation](#) help topic for more information.

[url](#) (of class [transfer window](#)) – The URL that refers to the server and remote folder currently displayed in a transfer window.

### New constants

[by security](#) – A new constant for the [view order](#) property of class [fetch shortcuts window](#) and class [shortcut list window](#). Specifies that the shortcut list window is sorted by the type of security the connection uses.

[SFTP](#) – A new constant for the [authentication](#) property of many classes and commands. Specifies the SFTP connection type. See the [SFTP](#) help topic for more information.

### Renamed classes

Fetch 4	Fetch 5
<a href="#">remote directory</a>	<a href="#">remote folder</a>
<a href="#">shortcut window</a>	<a href="#">fetch shortcuts window</a>

### Renamed commands

Fetch 4	Fetch 5
<a href="#">edit</a> (in Interarchy Suite)	<a href="#">interarchy edit</a>
<a href="#">view file list</a>	<a href="#">view verbose file list</a>

### Renamed properties

Of class `Application` :

Fetch 4	Fetch 5
<code>add IC suffix</code>	<code>add other suffix</code>
<code>blink icon</code>	<code>bounce icon in dock</code>
<code>default upload binary format</code>	<code>default upload format</code>
<code>proxy userid</code>	<code>proxy username</code>
<code>text mode edit with transfers</code>	<code>use text mode transfers for editing</code>
<code>warn on cleartext to secure server</code>	<code>warn on insecure password to Kerberos server</code>

Of class `transfer window` :

Fetch 4	Fetch 5
<code>bytes transfered (sic)</code>	<code>bytes transferred</code>
<code>current directory</code>	<code>current folder</code>
<code>initial directory</code>	<code>initial folder</code>
<code>parent directory</code>	<code>parent folder</code>
<code>userid</code>	<code>username</code>

Of class `shortcut` :

Fetch 4	Fetch 5
<code>userid</code>	<code>username</code>

## Renamed constants

Constants for `view order` properties:

Fetch 4	Fetch 5
<code>byName</code>	<code>by name</code>
<code>byDate</code>	<code>by date</code>
<code>bySize</code>	<code>by size</code>
<code>byNameWithCase</code>	<code>by case sensitive name</code>
<code>byHost</code>	<code>by host</code>
<code>byUserid</code>	<code>by username</code>
<code>byPath</code>	<code>by path</code>
<code>byKind</code>	<code>by kind</code>

Constants for `text line ending style` property of class `Application`:

Fetch 4	Fetch 5
<code>Mac style</code>	<code>Mac Classic style</code>
<code>UNIX style</code>	<code>Mac OS X style</code>

Constants for `background notification` property of class `Application`:

Fetch 4	Fetch 5
<code>BEEP Notification</code>	<code>Alert Sound Notification</code>
<code>Sound Notification</code>	<code>Custom Sound Notification</code>
<code>Speak Notificaton (sic)</code>	<code>Speak Notification</code>

Constant for various [format](#) properties:

Fetch 4	Fetch 5
<a href="#">Raw Data</a>	<a href="#">Binary</a>

## Changes to properties

[version](#) (of class [Application](#)) – Now returns [plain text](#) instead of [integer](#).

## Removed classes

[help window](#) – The help window no longer exists in Fetch 5, having been superseded by the Apple Help-based help.

## Removed properties

[automatic defaults to text](#) (of class [Application](#)) – Fetch 5's handling of download modes has changed significantly, making this property obsolete. See the [Transfer Options](#) help topic for more information.

[default binary file type](#) (of class [Application](#)) – Fetch 5's handling of file types has changed significantly, making this property obsolete. See the [Transfer Options](#) help topic for more information.

[default binary file creator](#) (of class [Application](#)) – Fetch 5's handling of file types has changed significantly, making this property obsolete. See the [Transfer Options](#) help topic for more information.

[dont use navigation services](#) (of class [Application](#)) – Fetch 5 (and Mac OS X) always use Navigation Services dialogs.

[text files creator](#) (of class [Application](#)) – Fetch 5's handling of file types has changed significantly, making this property obsolete. See the [Transfer Options](#) help topic for more information.

[translate iso](#) (of class [Application](#)) – The Translate ISO characters preference of Fetch 4 has been replaced by a more comprehensive character encoding preference in Fetch 5, which can be referenced using the [preferred encoding](#) property instead. See the [Character Encoding & Translation](#) help topic for more information.

[use internet config](#) (of class [Application](#)) – Since there is no longer an Internet Config control panel on Mac OS X, Fetch now reads settings from Internet Config but doesn't save changes back to it.

[including text files](#) (of class [transfer window](#)) – Fetch 5's handling of upload formats has changed and this property is obsolete. See the [upload formats](#) help topic for more information.

[modification date](#) (of class [shortcut](#)) – Fetch 5 no longer tracks the modification date of a shortcut.

[size](#) (of class [shortcut](#)) – Fetch 5 no longer explicitly tracks the size of a shortcut, although you can get the size of the objects referenced by a shortcut by using the [get information about](#) command.



## [Fetch Help](#) > [Using Fetch](#) > What is Fetch?

Fetch is a Macintosh program that makes it easy to transfer files — such as webpage, text, and image files — between your Macintosh and other computers over networks such as the Internet.

Typical uses of Fetch are:

- Publishing and maintaining a website at a web hosting provider
- Transferring documents to a printing company, service bureau, or clients you are working with
- Submitting advertisements or articles to newspapers and magazines
- Moving files between your Macintosh and Macintosh, UNIX, Windows, or other sorts of computers, as well as specialized equipment including medical imaging machines, image editing systems, and photo processing equipment

To transfer files via the Internet, you need two things: a client program on your Macintosh, and permission to access another computer, referred to as a server. You use the client program to connect to the server, and then you can either transfer files to the server (called uploading), or request that files be transferred back to your Macintosh (called downloading).

The client and the server use a special language and set of rules, called a protocol, for communicating and transferring files between the two computers. FTP (File Transfer Protocol) is one of the oldest and most established ways of moving files between two computers on the Internet.

Fetch is an FTP client for your Macintosh, that transfers files to and from FTP servers.

Fetch also supports SFTP and FTP with TLS/SSL, which are newer, more secure file transfer protocols.

Fetch has features that make it easy to update and maintain your files on the server. For example, you can:

- Quickly access frequently used servers by creating shortcuts
- Rename, move, delete, and change the permissions of files on the server
- Edit files on the server and have those changes uploaded back to the server automatically
- Upload only new or changed files with the Mirror command
- Create a droplet shortcut that lets you drag files to its icon to automatically connect and start uploading them
- Automate tasks using Automator and AppleScript

Fetch can transfer any kind of file, including text, HTML, image, audio, movie, Flash, Microsoft Office, and compressed archive files. You can use Fetch to transfer files to and from multiple different servers.

Normally your hosting company or the organization you are working with have already set up an FTP server, and you just need Fetch to communicate with it; you do not have to set up your own server. Neither Fetch the program nor Fetch Softworks the company provide a service or server for storing files. If you need a place to store your website or files, you should contact a hosting provider, and they can help set up an account on a server for you. See the [What do I enter in the hostname, username, and password fields?](#) topic for more information.

Fetch's goal is to be an easy-to-use, friendly, and reliable FTP client that you can use transferring your files with servers on the Internet. For more information on using Fetch, read the [quick start guide](#) or the [introductory tutorial](#), see the [list of how-to topics](#), browse the [Reference](#) help section, or use the Help search box to search for topics.

For information about licensing and buying Fetch, see the [Licensing Fetch](#) help topic.



If you are not familiar with FTP or Fetch at all, you may want to read the [What is Fetch?](#) and [Terms You Should Know](#) help topics to familiarize yourself with terminology commonly used in Fetch and the Fetch Help.

You may also want to read the [introductory tutorial](#), which has illustrated steps that walk you through the process of connecting to a server, navigating around its folders, and transferring a file to and from your Macintosh.

## Connecting

When you start Fetch or choose the **File > New Connection** command, the New Connection dialog appears. You must specify the hostname or IP address of the server you want to connect to (the "host"), either by typing it, or by choosing an item from the Recent Connections  or Shortcuts  pop-up menus to the right of the hostname field. If required, enter a username and password. Then click the **Connect** button. In order to use "anonymous FTP," you only need to enter the hostname; Fetch will automatically send the username "anonymous" and a default password. To save this information for easy access later on, choose **Make Shortcut** from the Shortcuts  pop-up menu before clicking the **Connect** button.

**Note:** The information entered in the New Connection dialog has nothing to do with your license to use Fetch. It specifies the server with which you are going to transfer files. If you are unsure of the hostname and your username, contact your Internet provider (the company that operates the server in question).

See the [New Connection dialog](#) help topic for more information.

## Folders (Directories)

Remote computers divide their files among "directories," which are equivalent to folders on the Macintosh. Fetch displays directories as folders, and in Fetch and the Fetch Help they are referred to as folders or "remote folders." Fetch displays remote folders similar to the way the Mac OS Finder displays folders in list view. Once you are connected, the name of the current folder is shown in the title bar of the transfer window. You can change to a different folder in several ways:

- Double-clicking the name of a folder in the file list (or selecting it and pressing Command-O or Command-Down arrow) will go to that folder.
- The **Path** button in the transfer window toolbar displays a pop-up menu that lets you go to any folder that contains the current folder.
- The **Back** button in the transfer window toolbar takes you back to the folder you were previously viewing.
- Choosing the **Go > Go to Folder** command prompts you for a folder name and, assuming you enter a valid folder name or path and have permission to view that folder, changes to that folder.
- Finally, the **Go** menu and the **Recent** button in the transfer window toolbar list folders you have recently visited on this server. You may quickly return to any of them by choosing one from the menu.

## Getting Files

To get (download) a file or folder, select the file and click the **Get** button in the toolbar. You can also get a file by double-clicking its icon or name.

To get multiple files or folders, click them while holding down the Command or Shift key, and then click the **Get** button in the toolbar.

You can also get files and folders by dragging them from the Fetch file list to the Finder.

To get a file correctly, Fetch has to communicate with the server in text mode or binary mode. Normally Fetch uses **Automatic** download mode, in which it decides the mode to use for a file based on its filename extension. However, if you know what kind of file you are getting and **Automatic** is not picking the correct mode, you can choose the appropriate mode — either **Text** or **Binary** — from the **Mode** submenu in the **Remote** menu.

By default, Fetch will decode any encoded files, such as BinHex, MacBinary, AppleSingle, or Stuffit files. See the [download modes](#) and [Transfer Options](#) help topics for more information.

For more information about getting files, see the [Get](#) command help topic.

## Putting Files

To put (upload) a single file from your Macintosh to another computer, click the **Put** button in the transfer window toolbar, or choose **Put** from the **Remote** menu. You will be prompted to select a file. You can also enter a new name to give to the uploaded file, or choose a different format for uploading the file. **Automatic** format is the recommended setting, as it will usually pick the correct format for the file. See the [upload formats](#) help topic for more information.

To put multiple files or folders, click the **Put** toolbar button, and then use the Shift and Command keys to select multiple files and folders in the file choice prompt, as you would in the Mac OS Finder.

You can also put files and folders on the remote computer by dragging them from the Finder to a Fetch file list.

For more information about putting files, see the [Put](#) command help topic.

## Stopping and Quitting

You may stop any ongoing action by clicking the **Stop**  button. Some servers do not recognize the stop command, and will continue the current action until it is completed anyway. In that case, your only options are to close the connection (by closing the transfer window), or quit Fetch by choosing the **File > Quit** command.

See the [Using Fetch](#) section of the help for more guided topics on working with Fetch, or browse the various help sections and use the Help search box to find information.



[Fetch Help](#) > [Using Fetch](#) > **Introductory Fetch Tutorial**

The following help items walk you through the process of connecting to a server, navigating around its folders, and transferring a file to and from your Macintosh. This tutorial assumes that you are familiar with the Macintosh. Some terms will be defined as we go along, others are defined in the [Terms You Should Know](#) help topic. If you are only interested in transferring webpages to a server, you can skip to the [Uploading Webpages](#) tutorial topic.

[Connecting](#)

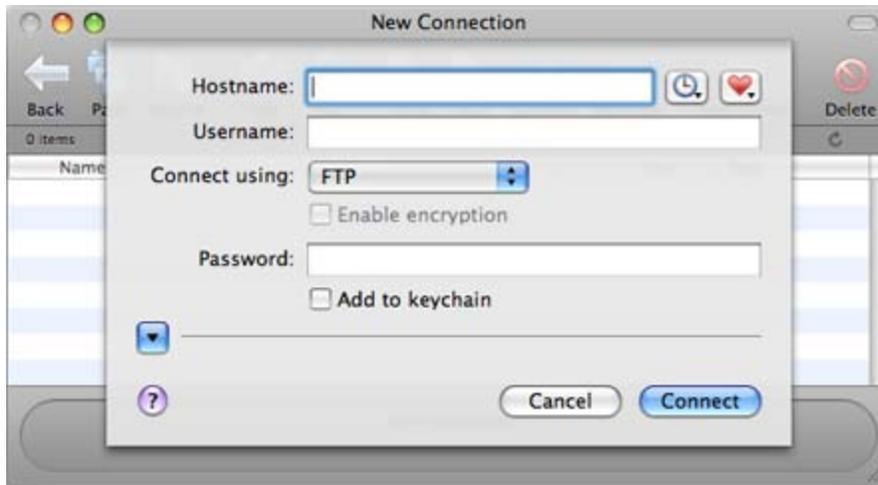
[Finding Files and Folders](#)

[Getting a File](#)

[Putting a File](#)

[Uploading Webpages](#)

When you first start Fetch, the New Connection dialog appears. You can choose **New Connection** from the **File** menu to display the dialog at other times.



The New Connection dialog has fields for entering the following pieces of information:

- Hostname: the name of the server you want to transfer files from or to
- Username: the name of your account on that server
- Connect using: how you're going to connect to the server (usually FTP or SFTP)
- Password: the password for your account

Usually you'll be entering information about an account you have with a service provider, such as your Internet service provider or a web hosting service. You get the hostname, username and password from that service provider. Fetch Softworks is not a service provider. For more information, see the [What do I enter in the hostname, username, and password fields?](#) topic. After you enter the information, you can save it in a shortcut so that you don't have to type it again in the future.

The New Connection dialog also has two ways to quickly fill out some or all of these fields. To the right of the Hostname field, there are two pop-up menus — the Recent Connections  pop-up menu and the Shortcuts  pop-up menu . The Recent Connections pop-up lists the last 10 connections you've made, whereas the Shortcuts menu list shortcuts to servers (similar to bookmarks in web browsers). The example below shows how to connect and create your own shortcuts.

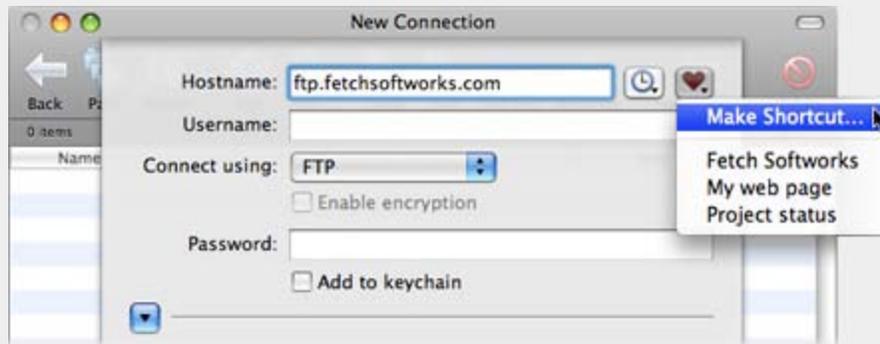
#### Example:

Choose **New Connection** from the **File** menu to display the New Connection dialog, if it's not already displayed.

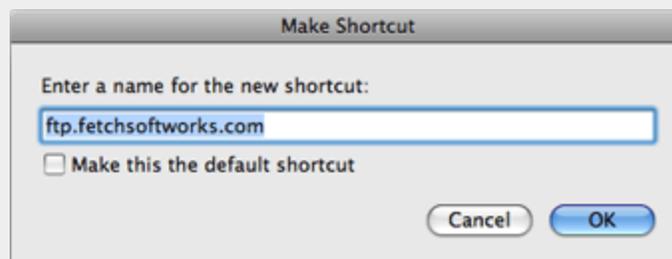
In the Hostname field, type "ftp.fetchsoftworks.com" (without the quotes).

Normally you would also type your account's username and password into their respective fields, but for this example you are using what's called "anonymous" FTP — you are connecting to ftp.fetchsoftworks.com as a guest with limited ability to make changes. No username or password is required for anonymous FTP.

Click the Shortcuts  pop-up menu, and choose **Make Shortcut**.



The Make Shortcut dialog appears.



You can either keep the suggested name, or enter a new one, such as "Tutorial Shortcut." If you wanted Fetch to always start with the information in this shortcut, you would check the **Make this the default shortcut** box. You probably don't want to make this tutorial shortcut the default, though, so don't check the box this time. Click the **OK** button to confirm the shortcut name.

Finally, click the **Connect** button to open the connection. The Fetch transfer window and file list will appear, and the shortcut will be saved for future use.

You do not have to make a shortcut each time you make a connection — usually you only make one if you think you will be connecting to the same server repeatedly.

When you want to make a connection, follow the steps in the example, but type in the information for your account and Internet provider instead, including your username and password. Whenever you want to use a shortcut you've created, choose it from the Shortcuts  pop-up menu in the New Connection dialog, or from the **Shortcuts** menu.

For more information about the New Connection dialog, see the [introductory](#) or [in-depth](#) help topics about it.

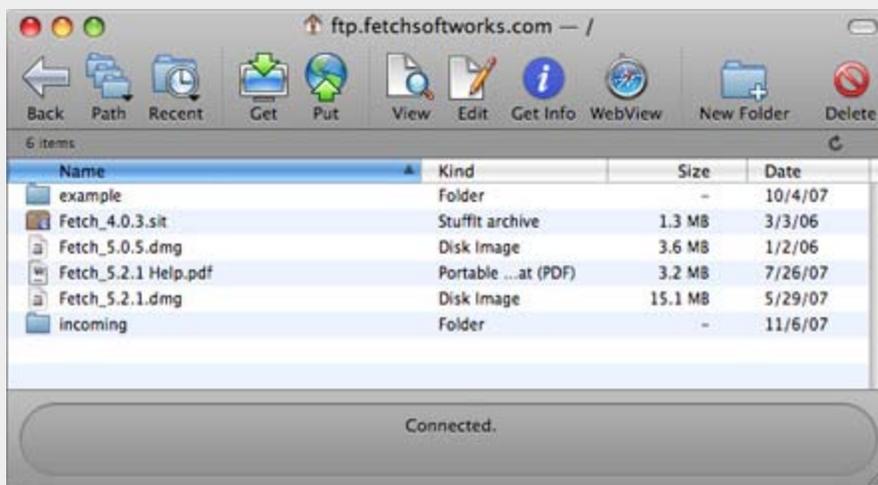
[Next](#)

## [Fetch Help](#) > [Fetch Tutorial](#) > Finding Files and Folders

Most servers organize their files among folders just like the Macintosh (although on other operating systems folders are often referred to as directories). Unless the file you want to download is in the first folder displayed by the server when you connect — which is usually either your home folder or the root folder (denoted by "/" ) — you'll have to navigate to the folder where the file is located. The name of the current folder is displayed in the title bar of the transfer window. The central part of the transfer window, the file list, contains the names of the files and subfolders for the current folder.

Example:

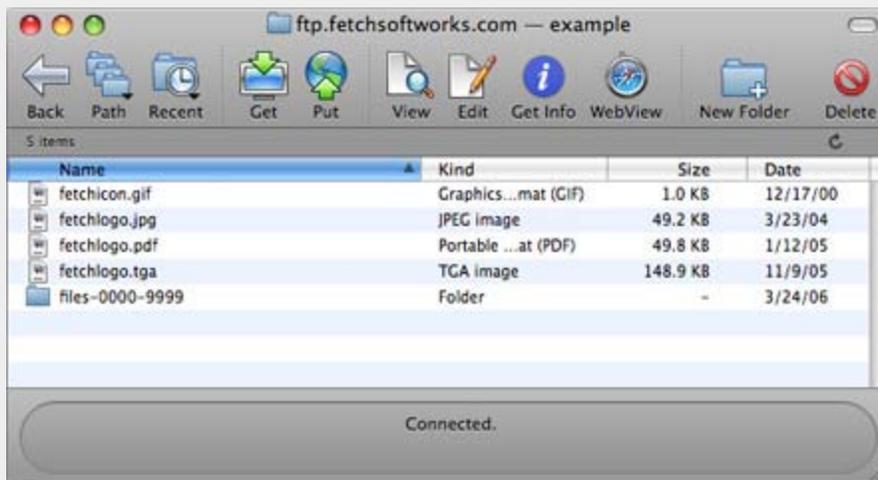
As in the [previous](#) step, open a connection to `ftp.fetchsoftworks.com`. You should see the transfer window. Your current folder will be "/", and you will see some folders in the file list.



For the purposes of this tutorial, we'll navigate to one of the folders.

You can change the current folder in several ways:

- The simplest is to double-click the name of a folder in the file list. You can try this by double-clicking the name "example" in the current file list. The current folder will become "example" and the file list will show the items in that folder.



- You can see folders you have recently visited in the **Go** menu, or by clicking the **Recent** button in the transfer window toolbar. Choose one of the folders listed to make it the current folder.



- If you click the **Path** button in the transfer window toolbar, a pop-up menu shows a list of folders containing the current folder (similar to clicking the title of a Finder window while pressing the Command key). Choose one of the folders listed to make it the current folder. (In the root folder, you won't see any other folders listed in the **Path** pop-up because you're at the top folder.)



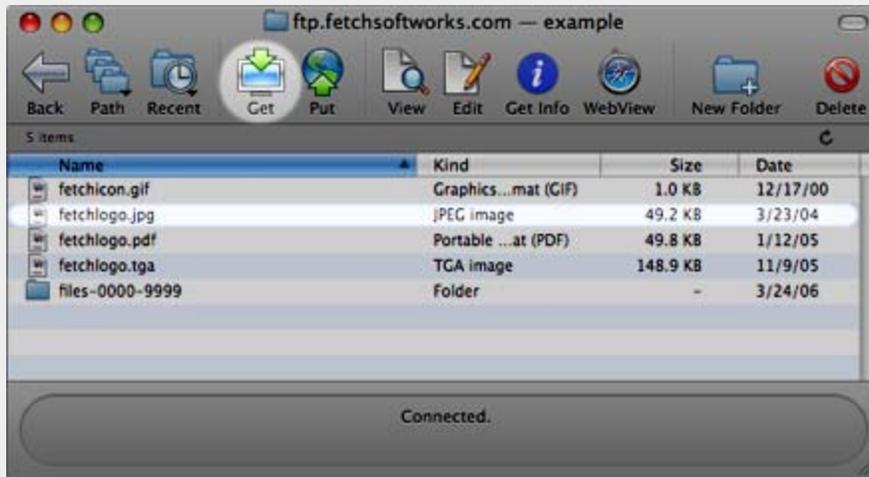
[Next](#)

 [Fetch Help](#) > [Fetch Tutorial](#) > Getting a File

The process of getting a file from an FTP or SFTP server is also known as downloading a file.

If you are not still connected to ftp.fetchsoftworks.com from the [previous](#) tutorial step, reconnect to it.

The current folder should be "example" (its complete path is "/example"). One of the files listed should be a JPEG image named "fetchlogo.jpg".



In order to transfer "fetchlogo.jpg" from ftp.fetchsoftworks.com to your Macintosh, select the file and click the **Get** button in the transfer window toolbar. Or you can double-click the file's name or icon, or select the file and press Command-Down arrow.

The file will be downloaded to the download folder specified in the [Download Preferences](#) pane. By default this is either your Downloads folder (Mac OS X 10.5 and later) or your Desktop folder (Mac OS X 10.4). If you do not have a download folder set, you will be prompted to name the file and choose a location to save it; after you click **Choose**, the file will be saved to your disk, ready to be used.

You can also drag the file from the Fetch transfer window to a folder or hard disk on the Macintosh desktop (or to the Desktop itself).

[Next](#)

## [Fetch Help](#) > [Fetch Tutorial](#) > Putting a File

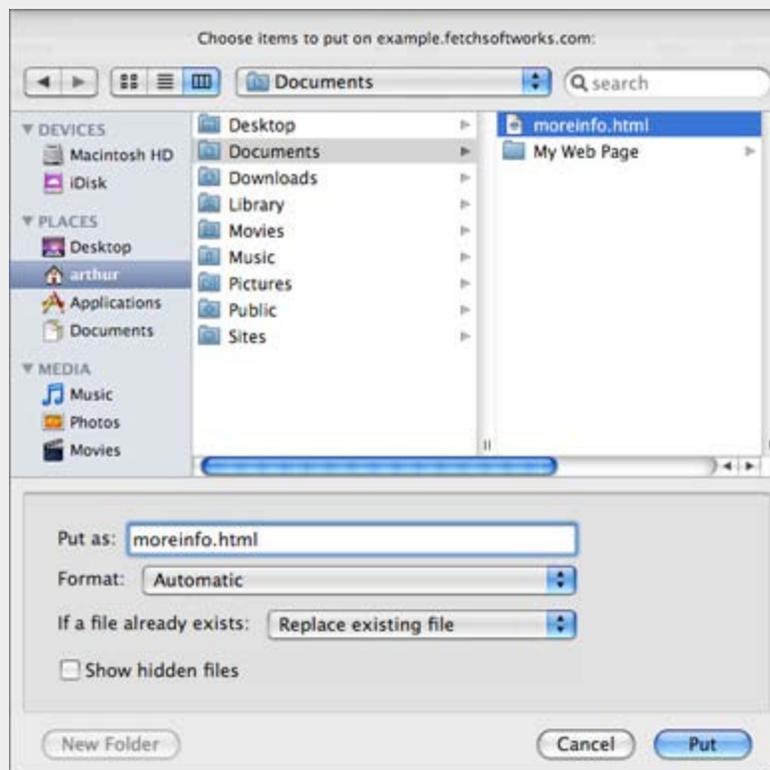
In order to put (upload) a file on an FTP or SFTP server, you must have write permission for a folder on the server (that is, the permission to create, rename, and delete files). Most anonymous FTP sites do not give guests write permission, so you will probably need to use a personal account for this part of the tutorial.

To upload a file to your FTP or SFTP server, first enter the hostname, username and password in the [New Connection dialog](#) and connect to the server.

Then click the **Put** button in the transfer window toolbar.



You will be prompted to select the file you want to transfer; since this is just a tutorial, you can pick any file, though you probably want to avoid very large files.



The file put prompt also asks you to provide a name to give the file on the server, and gives you a chance to pick an upload format from a pop-up menu. Fetch almost always picks an appropriate name and format so you can simply click the **Put** button.

You can select multiple files and folders to upload by holding down the Shift or Command keys when clicking on items, as you would in the Mac OS Finder.

If you can see the Finder icon for the file you want to upload, another way to upload it is to

simply drag the file's icon to the Fetch transfer window. In that case, Fetch will use the upload format specified by the **Remote > Format** submenu and name the file automatically.

The file will be transferred to the server and you will be able to watch its progress in the status pane at the bottom of the Fetch transfer window. When the transfer is complete, the name of the uploaded file will appear in the file list. If you do not want to leave the file there, select it and click the **Delete** button in the toolbar (or press Command-Delete).

Congratulations — you have just transferred files to and from your Macintosh! You can learn more about Fetch by consulting the [Using Fetch](#) and [Reference](#) sections of the help.

[Next](#)



One common use for Fetch is uploading files to web servers. Web authors often create their HTML, GIF, JPEG, etc. files on a Macintosh, and use Fetch to transfer the files to the computer that will actually serve the files over the Internet. Adding an image file to an eBay item description is similar: the image is first uploaded to a web server, and then a reference to the image is added to the item description. (See the [Uploading pictures for eBay](#) help topic for more information.) This tutorial topic walks you through your first such transfer. If you want more information, you can read the other tutorial topics, but they are not necessary for following along with this topic.

If you are using iWeb to create your website, you should also read the [Using Fetch with iWeb](#) help topic for instructions and tips specific to iWeb.

Before you start, you need to gather three important pieces of information:

1. The hostname of the server that will hold your pages. It should be something like "www.example.com". You don't want an entire address (such as "http://www.example.com"), just the hostname part.
2. Your username, that is, the name which you use to sign on to the server. Your username is usually a sequence of letters or numbers (such as "jsmith"), and may be the characters to the left of the "@" in your email address. It should not be confused with an email address, as your username usually won't include an @ sign.
3. Your password. Again, this may be the same password you use for email or dialing in to your service provider.

If you do not have these pieces of information, the company that sells you the web server space (or the university that provides it for free) should be able to tell them to you.

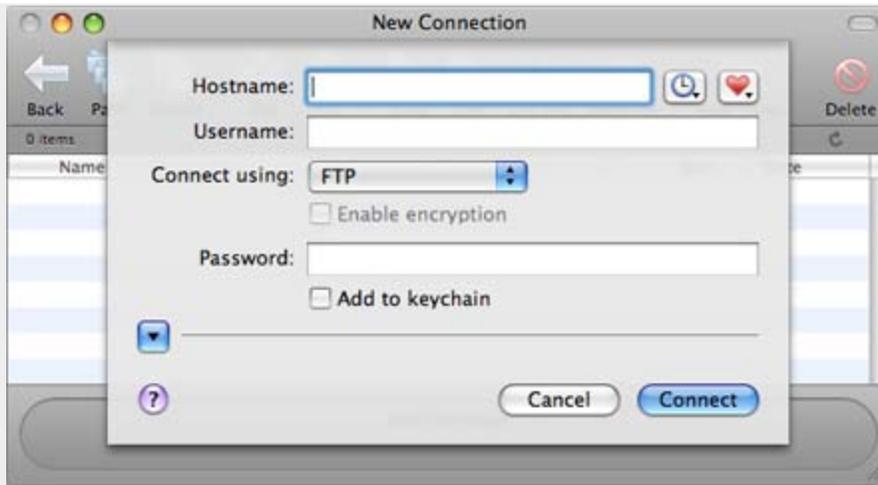
You should also find out if your server uses FTP or the newer, more secure SFTP or FTP with TLS/SSL for connections. If SFTP or FTP with TLS/SSL is available you should use one of them, but not all servers support them yet.

To verify your website's contents after you've uploaded the files, you'll need to know the address (URL) that refers to your site. Once again, you can get this information from your service provider.

If you are creating and uploading a website with multiple files, we recommend that you keep all the files in the same folder on your Macintosh. Some HTML editing programs put images in a separate folder, but this makes it more difficult to keep the links between files correct. Moving files to a new folder will probably break some links and image tags; you should test to make sure they all work before uploading.

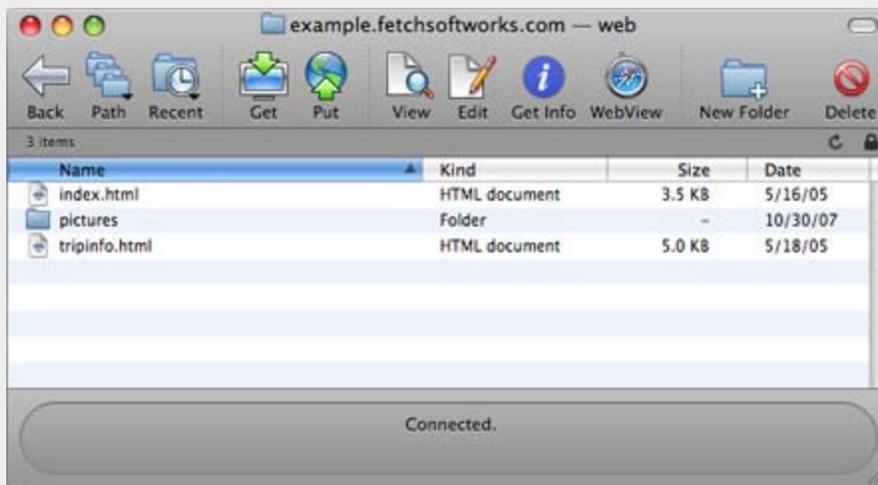
To start:

- Choose the **New Connection** command from the **File** menu.



- Enter your hostname, username, and password in the appropriate text fields of the New Connection dialog.
- Possibly change the **Connect using** pop-up menu to SFTP or FTP with TLS/SSL if your server supports secure connections.
- Click the **Connect** button.

You should see a list of the files that make up your website.



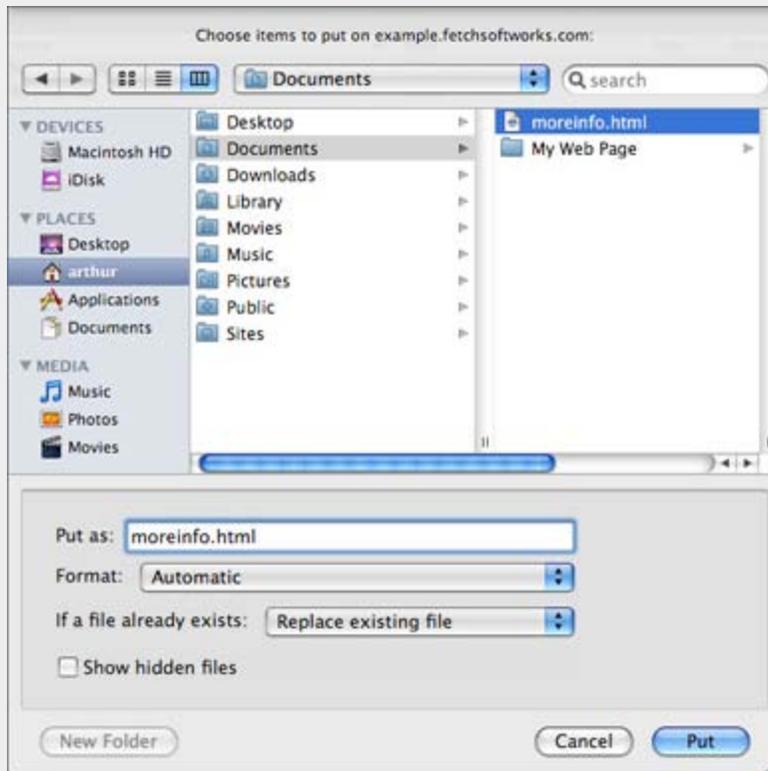
Otherwise, you will need to correct one or more of the pieces of information you entered in the New Connection dialog.

If you've successfully connected to the server and are viewing the folder with your web files, you can choose **New Shortcut** from the **Shortcuts** menu to create a shortcut for future use. This shortcut will be added to the Fetch Shortcuts list; to use it later, choose it from the Shortcuts pop-up menu in the New Connection dialog (or choose it from the list of shortcuts in the **Shortcuts** menu).

Once you're connected, click the **Put** button in the toolbar or choose the **Put** command from the **Remote** menu to upload your web files.



A dialog for selecting files appears.



If all of the files are in one folder, you can navigate to that folder and select them all to upload. Select multiple files by holding down the Shift or Command keys when clicking. Otherwise, you can choose them one by one. You can also select folders. The dialog also offers several other options, such as what format to use for the transfer; the default choices of **Automatic** format and **Replace existing files** are usually what you want. Then click the **Put** button to begin the transfer.

Once the pages have been uploaded, you should verify them with a web browser, such as Safari, Netscape, or Internet Explorer. Enter your website's address in the browser's Address or Location field and check to make sure that all the images display correctly and that all the links work. If there are problems, double-check your HTML tags to make sure that they have the correct file references. Also, be aware of the particular requirements of your service provider. Some require that your main page be named "index.html", and some require uploading everything to a remote folder named "web" or "public\_html". CGI scripts may need to go in a particular remote folder, and you may need to set their execute permission bit — see the [viewing and setting permissions](#) help topic.

After you've uploaded your webpages successfully, you may want to set up WebView, which lets you select files in Fetch and click the **WebView** button to view them in your web browser, and use the [Copy Web Address](#) command to copy the web address of those files. See the [setting up WebView](#) topic for more information.

End of tutorial — see [Using Fetch](#) for more guided topics



For information on common Fetch tasks and questions, select a topic from the categorized list below. If you don't see what you're looking for, use the Search field above.

[What is Fetch?](#)

[Introductory Fetch Tutorial](#)

[Frequently asked questions \(FAQ\)](#)

---

### **Connecting**

[Tutorial: Connecting](#)

[What do I enter in the hostname, username, and password fields?](#)

[Making Fetch start with information for your server or website](#)

[Making secure connections](#)

[Solving problems with connecting](#)

### **Uploading**

[Tutorial: Putting \(uploading\) a file](#)

[Resuming an interrupted upload](#)

[Creating a droplet shortcut](#)

[Uploading pictures for eBay](#)

[Uploading files for specific people](#)

[Uploading using copy and paste](#)

[Using the Fetch widget for uploading](#)

[Why is uploading slower than downloading?](#)

### **Downloading**

[Tutorial: Getting \(downloading\) a file](#)

[Resuming an interrupted download](#)

[Changing the application a file opens with](#)

[Opening files automatically after downloading](#)

[Decoding files automatically](#)

[Saving original archive files after expanding them](#)

### **Webpages & iWeb**

[Tutorial: Uploading webpages](#)

[Using Fetch with iWeb](#)

[Setting up WebView](#)

[Solving problems with websites and iWeb](#)

### **Shortcuts (bookmarks)**

[Creating a shortcut](#)

[Deleting or changing a shortcut](#)

[Creating a droplet shortcut](#)

[Moving shortcuts to another Macintosh](#)

[Backing up your shortcuts](#)

### **Mirroring (synchronizing)**

[Mirroring a local folder to a remote folder](#)

[About the Mirror window](#)

## **Editing**

[Editing a file](#)

[Changing the editor for a file](#)

[Editing files automatically instead of downloading](#)

[Fixing problems choosing editors and helper applications](#)

## **Moving or copying files**

[Moving files and folders](#)

[Copying files using copy and paste](#)

## **Renaming files**

[Renaming files and folders](#)

## **Changing folders**

[Tutorial: Finding Files and Folders](#)

[Changing folders](#)

## **Permissions**

[Viewing and setting permissions](#)

## **Droplets**

[Creating a droplet shortcut](#)

## **Stuffit & Zip (Decoding & Expanding)**

[Decoding files automatically](#)

[Saving original archive files after expanding them](#)

## **Automating & Scripting**

[Scheduling automatic transfers with iCal](#)

[Introduction to using Automator with Fetch](#)

[More examples of using Automator with Fetch](#)

[Setting up a watch folder](#)

[Scripting Fetch](#)

## **Tips & Tricks**

[Time-saving ways to use Fetch](#)

[Tips & tricks](#)

## **Troubleshooting**

[Troubleshooting FAQs](#)

[Getting Support](#)

[Sending a transcript to Fetch Softworks](#)

## **Advanced topics**

[Allowing Fetch to work with the Mac OS X firewall \(Mac OS X 10.5 or later\)](#)

[Converting line endings when uploading with SFTP](#)

[Displaying VMS version numbers in file lists](#)

[Fixing problems choosing editors and helper applications](#)

[Importing Interarchy bookmarks](#)

[Importing Transmit favorites](#)

[Moving keychains and passwords to another Macintosh](#)

[Using SSH public/private keys or ssh-agent with Fetch](#)



## [Fetch Help](#) > [Using Fetch](#) > What do I enter in the hostname, username, and password fields?

When you first open Fetch you are presented with the New Connection dialog. It requests three pieces of information: a hostname, a username, and a password.

Fetch is a program for transferring files between your Macintosh and another computer. The other computer is referred to as a server. When you want to use Fetch to transfer files to someone, the person or organization must be running a server that receives the files; Fetch connects to that server over the Internet and then you can transfer the files to it (or from it, as the case may be).

Each file server has an address on the Internet — just like websites have addresses — and it's the server's address that you put in the hostname field. Most servers require that you enter a username and password, in order to prove that you have been authorized to transfer files to it. So the hostname, username, and password are what you need in order to connect to a server.

If you've already signed up with a service provider or hosting company to store your website or files, they should be able to tell you the hostname of the server, the username you need to connect with, and your password. They may have sent you this information when you signed up for their service.

If you're trying to use Fetch to transfer files to a server at your company or university, your network administrator or computing helpdesk should be able to tell you what to enter.

If you're trying to connect to another company's server, you should contact them to find out the hostname, username, and password you need to use.

If you need a place to store your website or files, you should contact a hosting provider (a company that provides and maintains server space for storing web pages, FTP files, etc.), and they can help set up an account for you.

Fetch Softworks is not a hosting provider. We do not provide a service, or servers, for storing your website or files. We just make Fetch, software that lets you move files from your Macintosh to servers run by other companies or people; therefore, we do not know your hostname, username, or password.

To have the New Connection dialog filled out automatically with your information, see the [making Fetch start with information for your server or website](#) help topic.

For more information about the New Connection dialog, see the [introductory](#) or [in-depth](#) help topics about it.



## [Fetch Help](#) > [Using Fetch](#) > Making Fetch start with information for your server or website

You can make Fetch automatically start up with the information for your favorite server, or the server containing your website, by setting up a default shortcut. A default shortcut will fill out the hostname and username fields every time the [New Connection dialog](#) is displayed, such as when you start Fetch.

To create a default shortcut:

1. Choose **File > New Connection**, if the New Connection dialog isn't already showing.
2. Enter the name of the server in the Hostname field.
3. If your connection requires a username and password, enter them in the appropriate fields.
4. If you want your password to be filled out automatically when you use this shortcut, check the **Add to keychain** checkbox.
5. Click the Shortcuts  pop-up menu, and choose **Make Shortcut**.
6. Enter a name for the shortcut. This can be anything that makes it easy for you to identify the shortcut later on.
7. Check the **Make this the default shortcut** checkbox.
8. Click the **OK** button.
9. Click **Connect** to connect and create the shortcut.

If you're not sure what information to enter in the New Connection dialog, see the [What do I enter in the hostname, username, and password fields?](#) help topic.

For more information about the fields and options in the New Connection dialog, see the [New Connection dialog in-depth](#) help topic.

If you want to make an existing shortcut the default, or change the default shortcut after you've created one, use the **Default shortcut** pop-up menu in the [General Preferences](#) pane.

If you would like Fetch to automatically connect to the server without needing to enter or confirm any information, you should create a [droplet shortcut](#).

Related topics:

- [Creating a shortcut](#)
- [What do I enter in the hostname, username, and password fields?](#)
- [New Connection dialog in-depth](#)



Fetch supports secure connections that will protect the privacy of your password and your data by encrypting them as they're sent over the network. Generally you should use secure connections if they are available (not all servers support secure connections).

To open a secure connection:

1. Choose **File > New Connection** .
2. Enter the hostname for the server and your username.
3. From the **Connect using** pop-up menu, choose **SFTP** or **FTP with TLS/SSL**, depending on which your server supports.
4. Enter your password.
5. Click **Connect** to open the connection.

If you receive an error when making the connection, the server you are trying to connect to may not support SFTP or FTP with TLS/SSL connections. Consult your network administrator or service provider.

You can also create shortcuts that use secure connections by choosing SFTP or FTP with TLS/SSL as the connection type when creating the shortcut.

Some organizations may support [Kerberos](#) for making secure connections in addition to, or instead of, SFTP or FTP with TLS/SSL. Choose **FTP with GSSAPI** from the **Connect using** menu in step 3 to use Kerberos instead, and omit your password, since your Kerberos tickets will be used instead. (**FTP with KClient** is an older type of Kerberos connection that generally should not be used if possible.)

Related topics:

- [Security](#)
- [SFTP](#)
- [FTP with TLS/SSL](#)



## [Fetch Help](#) > [Using Fetch](#) > Solving problems with connecting

Using the suggestions below, you can solve some of the most common problems you may have connecting to a server; if they don't help, contact Fetch support for further assistance.

1. [I don't know or I've forgotten my hostname, username, or password.](#)
2. [I received an error saying "login incorrect", "rejected password," or "server cannot be found," but I know that I entered the correct hostname, username, and password.](#)
3. [I can connect successfully, but the file list never appears, or I get an error after connecting.](#)

*I don't know or I've forgotten my hostname, username, or password.*

You should contact the company or person that operates the server to which you are trying to connect, and verify the hostname, username, and password you should use. If you are trying to upload webpages, usually this means you should contact the web hosting provider that you have signed up with or your Internet service provider. See the [What do I enter in the hostname, username, and password fields?](#) help topic for more information.

Fetch Softworks is not a hosting provider — we do not run or maintain servers for customers, so we do not have access to the information you need to use to connect. We just make Fetch, software that lets you move files from your Macintosh to servers run by other companies or people.

*I received an error saying "login incorrect", "rejected password," or "server cannot be found," but I know that I entered the correct hostname, username, and password.*

If you receive an error like this when trying to connect to a server that you've previously connected to successfully, usually this indicates a problem with the server. You should contact the company or person who runs the server to have them look into the problem.

If they confirm that there is not a problem with the server, verify your hostname, username, and password with them, and then double-check what you've entered in the [New Connection dialog](#) or the shortcut.

You may receive an error saying your password is incorrect even if the real problem is that you entered the incorrect hostname or username, or the server is experiencing problems.

Fetch is not involved in deciding whether your password is correct or not; Fetch sends what you type as your password to the server, and either the server says the password is okay, and allows Fetch to connect; or the server rejects the password and sends back an error message, which Fetch then reports to you.

*I can connect successfully, but the file list never appears, or I get an error after connecting.*

There are several things to try to resolve this problem.

First, toggle the **Use passive mode transfers (PASV)** preference:

1. Go to the **Fetch** menu and choose **Preferences**.
2. Click the **General** tab.
3. If the **Use passive mode transfers (PASV)** checkbox is checked, uncheck

- it; if it is checked, uncheck it.
4. Try connecting again.

If that doesn't help, next try turning off the Mac OS X firewall temporarily if it is on. Follow the appropriate directions below, depending on which version of Mac OS X you are using.

### Mac OS X 10.6

1. Open the Mac OS X **System Preferences**.
2. Click the **Sharing** icon.
3. Click the **Firewall** tab.
4. If the preferences say "Firewall On," click the **Stop** button. (You may need to first click the lock icon in the lower left of the window and enter your Mac OS X password to make this change.)
5. Try connecting again.

### Mac OS X 10.5

1. Open the Mac OS X **System Preferences**.
2. Click the **Security** icon.
3. Click the **Firewall** tab.
4. If the **Allow only essential services** button is selected, select **Set access for specific services and applications** instead.
5. Try connecting again.
6. If that does not help, return to the Firewall preferences, and select the **Allow all incoming connections** button.
7. Try connecting again.

### Mac OS X 10.4

1. Open the Mac OS X **System Preferences**.
2. Click the **Sharing** icon.
3. Click the **Firewall** tab.
4. If the preferences say "Firewall On," click the **Stop** button.
5. Try connecting again.

If you find the only way to connect successfully is to change your firewall settings, you may want to contact Fetch support for further advice.

If that doesn't help, or the firewall was already off, finally try toggling the **Use 'LIST -al' command to reveal hidden files** preference:

1. Go to the **Fetch** menu and choose **Preferences**.
2. Click the **Miscellaneous** tab.
3. If the **Use 'LIST -al' command to reveal hidden files** checkbox is

checked, uncheck it; if it is checked, uncheck it.  
4. Try connecting again.

See the following help topics for more information about connecting in Fetch:

- [Tutorial: Connecting](#)
- [New Connection dialog introduction](#)
- [Making Fetch start with information for your server or website](#)



While Fetch tries to detect stalls and network or server errors and automatically resume uploads for you, if you accidentally stop the upload of a large file before it completes (for example by quitting Fetch), it is sometimes possible to save time by resuming the upload of the file without re-uploading the part already on the server.

You can only manually resume uploads for binary files that are not encoded when they're uploaded — usually these are graphic files (such as JPEG or TIFF), PDF files, sound files (such as AIFF or MP3), or movie (QuickTime) files. You generally cannot resume the upload of text files. When you cannot resume the upload of a file, the resume upload option in the Put dialog will be disabled.

To resume the upload of a file:

1. Open a connection to the server where the partial file is.
2. In the transfer window, navigate to the remote folder where the partial file is, if necessary.
3. Choose **Remote** > **Put** to open the Put dialog.
4. In the Put dialog, navigate to the local folder containing the file you want to resume uploading, and select the file.
5. From the **If a file already exists** pop-up menu, choose **Resume upload of file**.
6. Click the **Put** button.
7. The upload will start from where it left off.

Unlike the [Resume Download](#) feature, Fetch does not keep track of interrupted uploads and does not present a list of them to you.

Note that Fetch makes no comparisons to make sure the file you are uploading is the same as the file on the server. As long as the names are the same and the local file is longer than the one on the server, Fetch will add the end of the local file to the end of the existing remote file. You should be sure the two files are the same before using this feature, otherwise you may end up with a file made up of parts of two different files. If the new file is shorter than the existing file, you will get an error.

You can resume uploads for any file uploaded in Binary (Raw Data) format, or in Automatic format where Automatic chooses Binary (Raw Data). This means you cannot resume the upload of text files, unless you were uploading them as Binary (Raw Data), nor can you resume the upload of files uploaded in BinHex, MacBinary, or AppleSingle, Zip Archive, or Stuffit X Archive formats.

Not all servers support resuming uploads. If your server does not, you will receive an error.

If you were uploading more than one file or a folder when the upload stopped, the manual resume upload feature can only finish uploading the file that was being transferred at the time — Fetch does not remember the list of files it was going to upload, but hadn't gotten to yet. You will have to upload any other un-uploaded files again manually. (However, Fetch's automatic resume upload feature does work to finish uploads of folders or multiple files.)

Related topics:

- [Put command](#)

- [Resuming an interrupted download](#)

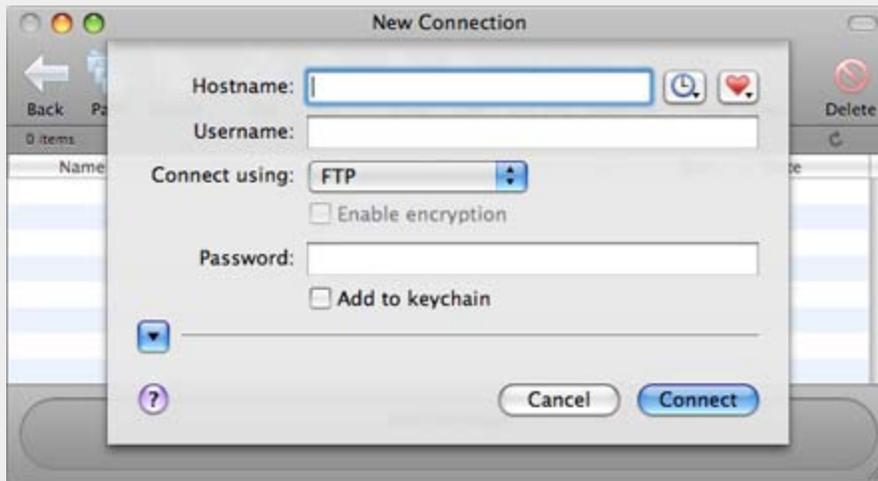
## [Fetch Help](#) > [Using Fetch](#) > **Uploading pictures for eBay**

A common use for Fetch is to upload pictures of items you're auctioning on eBay to a web server. Fetch is only part of what you need to accomplish this task—you also need an account with a service that offers web hosting, such as your Internet service provider or a web hosting service. That provider should be able to tell you the hostname, username and password you need to use in Fetch.

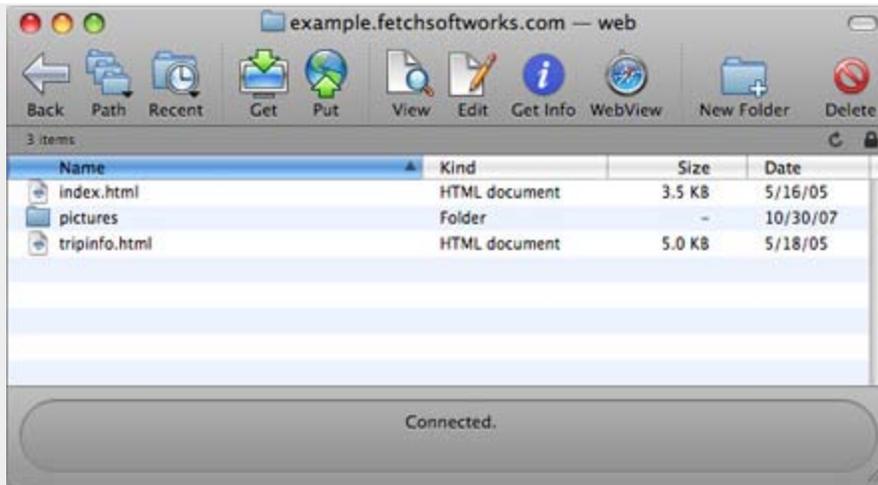
Fetch can only help transfer pictures from your Macintosh to a web server; it cannot host pictures for you. Fetch Softworks is not a service provider.

To upload pictures for use on eBay, follow these steps:

1. Create a digital picture of your item. Use a digital camera, scan the item with a scanner, or take your film to a store that can put your photos on CD.
2. Choose the **New Connection** command from the **File** menu. The New Connection dialog appears:



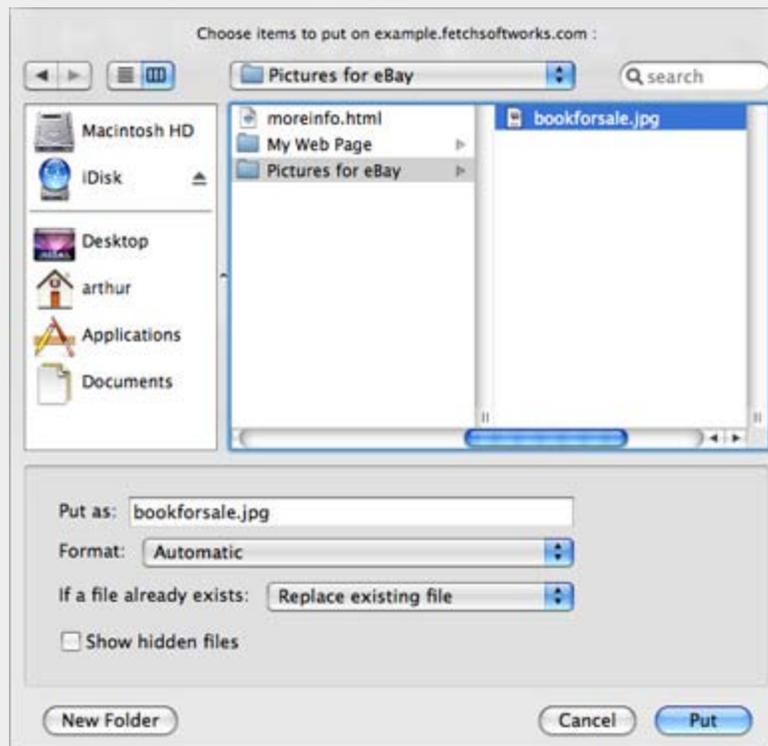
3. Enter the hostname of your server, your username, and your password in the appropriate text fields of the New Connection dialog. If you are unsure what to enter here, see the [What do I enter in the hostname, username, and password fields?](#) help topic.
4. Click the **Connect** button.
5. You should see a list of the files that make up your website:



- Once you're connected, click the **Put** button in the toolbar or choose the **Put** command from the **Remote** menu.



- A dialog for selecting files appears:



- Navigate to the folder on your Macintosh where the picture of your item is stored.
- Select it.
- Click the **Put** button to begin the transfer.
- Once the file is transferred, you are done using Fetch for now.
- Go to eBay in your web browser and fill out the form for listing your item for sale. One

of the steps is to enter the web address (URL) for the picture you have just uploaded. The web address (URL) will look something like this:  
http://www.host.com/yourfolder/photo.jpg . The web address may be different from the hostname you used to upload the picture. Your service provider can tell you what the correct web address for your picture is. If you think you know it, you can test it by typing it into a new window in your web browser. If your picture appears, you're all set and you can enter that same address into the eBay form.

You can upload multiple pictures and link to them in your item description, in addition to the one you specify when you create the listing.

If you regularly upload pictures to a website for eBay, you may want to set up **WebView**, which lets you select files in **Fetch** and click the **WebView** button to view them in your web browser, and use the [Copy Web Address](#) command to copy the web address of those files. See the [setting up WebView](#) topic for more information.



## [Fetch Help](#) > [Using Fetch](#) > **Uploading files for specific people**

You can use Fetch to upload (put) files for other people, such as co-workers or customers, to download. If you want to make sure that only the intended recipients can read the files, follow these suggestions:

1. Encrypt the files on your Macintosh. One way to do this is to use [StuffIt](#) or another encryption program to create a password protected archive of your files.
2. Choose **File > New Connection**.
3. Enter the connection information for your FTP space or web space, and click the **Connect** button. If you are unsure what to enter in the New Connection dialog, see the [What do I enter in the hostname, username, and password fields?](#) help topic.
4. Click the **Put** toolbar button, or choose **Remote > Put**.
5. Select the encrypted files you created in step 1.
6. Click the **Put** button.
7. After the files have finished uploading, send a link (URL) for the files to the people who the files are intended for, and tell them the password to unlock the files. If you are unsure what the URL for the files would be, your Internet provider can help you.

This way, you don't have to worry that anyone could download the files, because only the intended recipients who know the password can decrypt them.

If you connect to the same server frequently, you may want to [create a shortcut](#) to it.

If you frequently send links to files that you upload, you may want to [set up WebView](#) for your server so that you can copy web addresses of files in Fetch.

While Fetch can create StuffIt archives when uploading files, it cannot create password protected StuffIt archives, so you must use the StuffIt application to create them. However, Fetch will prompt for the password to decrypt password protected StuffIt archives when downloading them.

Another way to limit access to files and folders on your server to specific people is to configure your web server to require a password to enter certain folders or download certain files. Fetch cannot do this for you; you would need to talk to your Internet service provider about how to set up your web account for password protection.



## [Fetch Help](#) > [Using Fetch](#) > **Uploading using copy and paste**

You can use the standard Macintosh Copy and Paste commands to upload files and folders in Fetch.

To do this:

1. In the Finder, select the files and folders you want to upload.
2. Choose **Edit > Copy**.
3. Open Fetch.
4. Connect to the server you want to upload the files to.
5. In the transfer window for the server, choose **Edit > Paste**.

Fetch will begin uploading your files.

Related topics:

- [Copying files using copy and paste](#)



## [Fetch Help](#) > [Using Fetch](#) > Using the Fetch widget for uploading

The [Fetch Dashboard widget](#) lets you easily upload files and folders to a specific server by dragging and dropping items on the widget. After the upload has started, the Fetch widget displays the progress of the transfer.

First, you must open a copy of the Fetch widget if one is not already open. You can also use these instructions to create another copy of the Fetch widget:

1. Open Dashboard, either by clicking the Dashboard icon in the Dock or by pressing the Dashboard keyboard shortcut (usually the F12 key).
2. Click the Open (+) button in the bottom-left of the screen to reveal the widget bar.
3. Use the arrow buttons in the widget bar to find the icon for the Fetch widget if necessary.
4. Click the Fetch widget icon or drag it out of the widget bar to open a new Fetch widget.
5. Click the Close (x) button to hide the widget bar again.

Before you can use the widget for uploading, you must configure it with a destination — either a shortcut or recent connection. If you've set a [default shortcut](#) in Fetch, the widget will be automatically configured with that shortcut. Otherwise, to set or change the widget's destination:

1. Move your mouse over the Fetch widget. A small Info (i) button will appear.
2. Click the Info (i) button to flip the widget to its back so you can configure it.
3. Click the Recent Connections  or Shortcuts  button to choose a destination for the widget. If the destination you want is not listed, you must [create a shortcut](#) in the Fetch application first, and then return to Dashboard.
4. Click the **Done** button to finish configuring the widget. The front of the widget will display the hostname of the server and the remote folder on that server that files will be uploaded to.

Now you're ready to use the widget for uploading. To upload files and folders:

1. In the Finder, select the files and folders you want to upload.
2. Click one of the items and start dragging.
3. With your other hand, press the Dashboard keyboard shortcut (usually the F12 key) to open Dashboard.
4. Continue dragging the files until they are over the Fetch widget. A copy  cursor will appear, and the name of the server will be highlighted when you have dragged the files to the correct place.
5. Drop the files by releasing the mouse button.
6. The widget will open a connection in Fetch and start uploading the files. Progress information will be displayed on the front of the widget.

You can open as many copies of the Fetch widget as you want, and configure each one with a

different shortcut or recent connection.

To change the keyboard shortcut for displaying Dashboard, open the **Dashboard & Exposé** pane in the **System Preferences**. (On Macintosh laptops that use the F12 key to eject a disc, Fn-F12 should open Dashboard, but you may want to change it to something more convenient.)

For more information, see the [Fetch widget](#) topic.

Related topics:

- [Droplet shortcuts](#)



## [Fetch Help](#) > [Using Fetch](#) > Why is uploading slower than downloading?

For many users, uploading files is quite a bit slower than downloading files. This is usually normal, because most high-speed Internet connections, including cable modems and DSL, are asymmetric — they are designed to provide much better speed for downloading than uploading. Since most users spend much more time downloading (which includes viewing web pages or multimedia files) than they do uploading, high speed Internet providers have designed their systems to give priority to downloading. So if your upload speed appears to be slower than your download speed, this is probably expected.

Another factor to be aware of is that providers advertise their speeds in kilobits, whereas Fetch reports speeds in kilobytes. 8 bits equal 1 byte, so the numbers you see in Fetch will appear to be smaller than the numbers advertised for your connection. You must multiply the number you see in Fetch by 8 for an accurate comparison. For instance, a 384/128 DSL connection is rated for 384 kilobits per second download, and 128 kilobits per second upload — which is equivalent to 48 kilobytes per second download and 16 kilobytes per second upload.

Your upload and download speeds will almost never match the maximum advertised speed of your connection. It's normal to only get 80–90% of the advertised maximum, and in the small print of your service provider's advertising you'll find that they only promise "up to" the advertised speed, not that you will always get that speed. This happens for several reasons. First, your connection may be shared with other people in your building or neighborhood, so if a lot of other people are using the Internet you may experience a slowdown. Second, servers may be busy uploading and downloading files for many different users, so they are unable to provide the maximum possible speed for your files. Finally, other network activity on your computer may slow down your transfer, since it must all share the same connection. If you try to transfer two files at the same time (in different transfer windows), each transfer will go slower because Fetch must split the network connection between them. Surfing the web or answering your email should only have a minor impact on transfer speeds.

Nevertheless, if you feel you are getting slower transfer speeds than normal, there are websites that you can use to test your connection speed. Your Internet service provider may have one, or you can try one such as the [BroadbandReports.com speed test](#). If you are repeatedly getting transfer speeds much lower than the ones reported by a test, you should try transferring your files to a different server to see if they go faster; and if so, contact your main server's administrator to see if there is a problem.



## [Fetch Help](#) > [Using Fetch](#) > Resuming an interrupted download

While Fetch tries to detect stalls and network or server errors and automatically resume downloads for you, if you accidentally stop the download of a large file before it completes (for example by quitting Fetch), it is often possible to save time by resuming the download of the file without re-downloading the part you already have:

1. Choose **Remote** > **Resume Download** to open the Resume Download window.
2. Select the name of the file you want to resume downloading, and then click the **Resume** button, or double-click its name.
3. Fetch will reopen the connection to the server the file is located on if necessary, and start the download from where it left off.

An interrupted transfer leaves a partial file in your download folder or the location you dragged the file to originally. You can also resume a download by double-clicking the partial file in the Finder.

Note that if you were downloading more than one file or a folder when the download stopped, the Resume Download feature can only finish downloading the file that was being transferred at the time — Fetch does not remember the list of files it was going to download, but hadn't gotten to yet. You will either have to go begin downloading any other untransferred files again manually, or see the advice below. (However, Fetch's automatic resume download feature does work to finish downloads of folders or multiple files.)

See the [Resume Download window](#) help topic for more information about using **Resume Download**.

### Resuming the download of multiple items

If you are trying to download many files and folders and your download was interrupted for some reason, you may want to consider using the [Mirror](#) command. The **Mirror** command lets you associate a folder on the server (the one with the files you're trying to download) with a folder on your Macintosh. Once you've done that, you can tell Fetch to mirror the contents of the remote folder to the local folder.

Fetch will compare the contents of the remote and local folders, and only download new or changed files. This means that files that have already been downloaded successfully won't be downloaded again. If your first attempt to mirror the folder stops before all the files are downloaded, you can tell Fetch to mirror again, and the next attempt will pick up where the last one left off, instead of starting from the beginning again. And after mirroring a few times, all your files will be downloaded.

Note that Fetch's automatic resume download feature should make this approach unnecessary under most circumstances.

See the [Mirror window](#) help topic for more information about mirroring.

Related topics:

- [Resuming uploads](#)



## [Fetch Help](#) > [Using Fetch](#) > Changing the application a file opens with

When you download files from a server to your Macintosh, normally each kind of file is assigned an application that it will open in. Fetch provides reasonable defaults based on the Mac OS settings for the common kinds of files, but also provides a way to change which application a specific kind of file will open in.

Fetch decides which application to assign to a downloaded file by its extension — the part of the filename at the end after a period, such as ".txt" or ".jpg" . The application a certain kind of file will open in is called the *preferred application* for that kind of file. Fetch also uses the preferred application to determine the appropriate icon to display for each kind of file in the [file list](#) of [transfer windows](#).

To set the preferred application for a kind of file, use the Transfer Options pane of the [info window](#):

1. Connect to your server, and find a file with the extension whose preferred application you want to change. You may need to upload an example file if you do not know where one is located.
2. Select the file.
3. Click the **Get Info** button, or choose **Remote > Get Info** .
4. Click the disclosure triangle next to the Transfer Options label at the bottom of the info window to display the Transfer Options pane, if it's not already displayed.
5. Click the pop-up menu labeled "Open files like this with"
6. Choose an application from the menu, or if the desired application is not listed, choose the **Other** item and select the desired application.

From now on, any downloaded file with the same extension as the file whose transfer options you just changed will open in the newly selected application when you double-click it in the Finder. Also, the icon displayed for that kind of file in the transfer window will update to reflect the new choice of application.

Note that changing the preferred application in Fetch only affects files downloaded by Fetch, not all files of that kind on your Macintosh. Use the Finder's Get Info command to change the applications all files will open with in the Finder. Also, changing the preferred application for opening the file does not change the application used for [editing](#) a file.

If files have been encoded in a special format that preserves Macintosh information, such as BinHex, MacBinary, AppleSingle, Zip Archive, or StuffIt X Archive, the decoded files will open in the application originally specified for those files when they were uploaded. If that setting was different on the computer that uploaded those files than on yours, they may not open in the application you expect.

You can also use the Transfer Options pane of the info window to tell Fetch to [open certain kinds of files automatically](#) after they've been downloaded, to [automatically edit files](#) instead of downloading them, and [change the application that does the editing](#). See the [Transfer Options](#) help topic for more information.

Related topics:

- [Get Info](#)
- [Info window](#)

- [Download Preferences](#)
- [File list](#)



## [Fetch Help](#) > [Using Fetch](#) > **Opening files automatically after downloading**

You can tell Fetch to automatically open certain kinds of files (such as text or HTML files) after downloading them in the Transfer Options pane of the [info window](#):

1. Connect to your server, and find a file whose name ends with an extension for the kind of file that you want to open automatically. For example, for webpage (HTML) files, find a file that ends in ".html".
2. Select the file.
3. Click the **Get Info** button, or choose **Remote > Get Info**.
4. Click the disclosure triangle next to the Transfer Options label at the bottom of the info window to display the Transfer Options pane, if it's not already displayed.
5. In the "After transferring a file like this" section, click the **Save and open the file** button.
6. Optionally you can also choose the application you want the file to open in (e.g. Dreamweaver) from the **Open files like this with** menu in the Transfer Options pane. If the application you want is not listed, you can choose **Other** to find it.

From now on, any time you download a file with the same extension (by double-clicking it or using the **Get** command), it will open automatically in the application you've chosen when the download is complete. You can set as many different kinds of files as you like to automatically open.

To stop automatically opening a kind of file, follow the instructions above except click the **Save the file** button in the "After transferring a file like this" section.

To stop files from decoding automatically after they are downloaded, uncheck the **Automatically decode files like this** checkbox. For more information, see the [decoding files automatically](#) help topic.

You can also use the Transfer Options pane of the info window to tell Fetch to [automatically edit files](#) instead of downloading them. Also see the [Transfer Options](#) help topic for more information.

Related topics:

- [Changing the application a file opens with](#)
- [Get Info](#)
- [Info window](#)

## [Fetch Help](#) > [Using Fetch](#) > **Decoding files automatically**

Normally Fetch automatically decodes certain kinds of encoded, compressed, and archived files (such as BinHex, MacBinary, Stuffit archives, or Zip archives) after downloading them. If that is the behavior you want, no changes are necessary.

However, you can tell Fetch to stop decoding all or certain kinds of encoded and archived files.

To tell Fetch to stop expanding or decoding all files, or re-enable automatic decoding of all files if you previously turned it off:

1. Choose **Fetch > Preferences** .
2. Click the **Download** tab.
3. Check or uncheck the **Allow automatic decoding of downloaded files** checkbox.

Note that disabling decoding of all files will disable Fetch's automatic decoding of BinHex, MacBinary, and AppleSingle files in addition to disabling Zip and Stuffit archive expanding.

To tell Fetch to stop automatically decoding a certain kind of file, or re-enable automatic decoding for that kind of file if you previously turned it off:

1. Connect to your server, and find a file whose name ends with an extension for the kind of file that you want to stop or start decoding automatically. For example, for Stuffit files, find a file that ends in ".sit", or for Zip files, find a file that ends in ".zip".
2. Select the file.
3. Choose **Remote > Get Info** .
4. Click the disclosure triangle next to the Transfer Options label at the bottom of the [info window](#) to display the Transfer Options pane, if it's not already displayed.
5. Check or uncheck the **Automatically decode files like this** checkbox.

Note that the **Automatically decode files like this checkbox** in the info window will be disabled if the **Allow automatic decoding of downloaded files** checkbox in the Download Preferences pane is unchecked.

When automatic decoding for a kind of file is on, any time you download a file with the same extension, Fetch will automatically decode or expand that file. When automatic decoding is off, Fetch will just save the original file. Automatic decoding may not apply if you have also selected the **Open files like this** or **Edit files like this** options for this kind of file in Transfer Options.

You can also use the Transfer Options pane of the info window to tell Fetch to [automatically open](#) or [automatically edit](#) certain kinds of files. Also see the [Transfer Options](#) help topic for more information.

Related topics:

- [Saving original archives files after expanding them](#)
- [Get Info](#)
- [Info window](#)
- [Stuffit and Archive formats](#)





## [Fetch Help](#) > [Using Fetch](#) > Saving original archive files after expanding them

Normally after downloading and decoding compressed or encoded files (such as Stuffit archives, Zip archives, or BinHex files), Fetch deletes the original archive files.

You can prevent this from happening, and keep the original archive files, for all or certain kinds of encoded and archived files.

To tell Fetch to stop expanding or decoding all files, and thus keep all archives, or re-enable automatic decoding of all files if you previously turned it off:

1. Choose **Fetch > Preferences** .
2. Click the **Download** tab.
3. Check or uncheck the **Allow automatic decoding of downloaded files** checkbox.

Note that disabling decoding of all files will disable Fetch's automatic decoding of BinHex, MacBinary, and AppleSingle files in addition to disabling Zip and Stuffit archive expanding.

To tell Fetch to stop automatically decoding a certain kind of file, and keep the original archive files, or re-enable automatic decoding for that kind of file if you previously turned it off:

1. Connect to your server, and find a file whose name ends with an extension for the kind of file for which you want to keep the original archives. For example, for Stuffit files, find a file that ends in ".sit", or for Zip files, find a file that ends in ".zip".
2. Select the file.
3. Choose **Remote > Get Info** .
4. Click the disclosure triangle next to the Transfer Options label at the bottom of the [info window](#) to display the Transfer Options pane, if it's not already displayed.
5. In the "After transferring a file like this" section, click the **Save and open the file** button.
6. Uncheck the **Automatically decode files like this** checkbox.
7. Choose Stuffit Expander from the **Open files like this with** menu if it's not already chosen.

Now when you download a compressed archive file, instead of Fetch expanding the file itself, it passes the file off to Stuffit Expander for expansion. Normally Stuffit Expander does not delete the file after expanding it.

If Stuffit Expander does not appear as an option in the **Open files like this with** menu, you may not have Stuffit Expander installed on your Macintosh, or you may need to upgrade your copy of Stuffit Expander. You can [download the latest free version of Stuffit Expander](#) and install it to solve this problem.

To have Fetch handle the decoding of the archived files after downloading again (and have them deleted automatically), follow the steps above, except select the **Save the file** button in the "After transferring a file like this" section and recheck the **Automatically decode files like this** checkbox.

Note that the **Automatically decode files like this checkbox** in the info window will be

disabled if the **Allow automatic decoding of downloaded files** checkbox in the Download Preferences pane is unchecked.

For more information about the other transfer options, see the [Transfer Options](#) help topic.

Related topics:

- [Get Info](#)
- [Info window](#)
- [Stuffit and Archive formats](#)



## [Fetch Help](#) > [Using Fetch](#) > Using Fetch with iWeb

If you're using Apple's iWeb to build a website, but you don't have a MobileMe (.Mac) account, you can use Fetch to upload your website to any web server. You'll find uploading your website is easy after you do it once.

The instructions are slightly different depending on which version of iWeb you have. Follow the appropriate directions below.

To publish your pages from iWeb 3.0 or later (iLife '09 or later) and upload them using Fetch:

1. Create a folder on your desktop with the name of your website (something like "My WebSite March 5").
2. Open iWeb.
3. Select your site by clicking its name in the sidebar on the left of the iWeb window.
4. In the Site Publishing Settings window, click the **Publish to** pop-up menu and choose **Local Folder**.
5. In the Folder Location section, click the **Choose** button.
6. You will be prompted to choose a folder to put your website files in. Navigate to and open the folder you created on your desktop, then click the **Choose** button.
7. Click the **Publish Site** button in the toolbar at the bottom of the window.
8. iWeb will export the files that make up your website.
9. Open Fetch.
10. Enter the information for your server into the New Connection dialog and click **Connect**.
11. A new window listing the folder and files on your server appears. This is the Fetch transfer window.
12. In the transfer window, open the folder where you want to put your website files. You can change to a different folder by double-clicking on its icon.
13. Open the website folder you created on your desktop. You will see an index.html and one or more folders.
14. Select all of these and drag them to the transfer window of Fetch.
15. Fetch will start uploading your files and folders. The dog cursor runs until Fetch has finished putting your files on to the server.
16. Once Fetch has finished uploading, visit your website to make sure it looks correct.

To publish your pages from iWeb 2.0.4 or earlier (iLife '08 or earlier) and upload them using Fetch:

1. Create a folder on your desktop with the name of your website (something like "My WebSite March 5").
2. In iWeb, go to the **File** menu, and choose **Publish to a Folder**.
3. You will be prompted to choose a folder to put your website files in. Choose the folder you created on your desktop.
4. iWeb will export the files that make up your website.
5. Open Fetch.

6. Enter the information for your server into the New Connection dialog and click **Connect**.
7. A new window listing the folder and files on your server appears. This is the Fetch transfer window.
8. In the transfer window, open the folder where you want to put your website files. You can change to a different folder by double-clicking on its icon.
9. Open the website folder you created on your desktop. You will see an index.html and one or more folders.
10. Select all of these and drag them to the transfer window of Fetch.
11. Fetch will start uploading your files and folders. The dog cursor runs until Fetch has finished putting your files on to the server.
12. Once Fetch has finished uploading, visit your website to make sure it looks correct.

If you are having problems with your website after following the instructions above, please see the [solving problems with websites and iWeb](#) help topic for solutions and troubleshooting advice.

If you haven't already, you may want to create a shortcut to your server for quicker access in the future. See the [creating a shortcut](#) help topic for more information.

WebView lets you set up a correspondence between files you are viewing in Fetch and the address of the webpage they are part of, so that you can easily view changes to your website in a web browser and copy the web addresses of those files.

Once you've set up WebView for a server, you can then click the **WebView** button in the toolbar of transfer windows to show the selected files or folders in your web browser. You can also use the [Copy Web Address](#) command to copy a web address (also called a link or URL) of the selected items in Fetch.

For more information about the concepts and features of WebView, see the in-depth [WebView](#) help topic.

Before you can use WebView, you must tell Fetch the web address of your server. (In most cases, Fetch cannot figure out your web server's address automatically; see below for an explanation.)

To set up WebView for a server:

1. Connect to your server using the [New Connection dialog](#) or by opening the shortcut for that server.
2. Go to your home folder, or the folder containing your webpages, if you are not already there.
3. Click the **WebView** button in the toolbar.
4. The WebView dialog should appear. (If it does not, and a webpage opens instead, WebView is already set up for this server — either you set it up already, or you are on a server whose information is built into Fetch already. In either case, you're done.)
5. Look at the server and folder name mentioned in the dialog, or displayed in the window's title bar. You are setting up WebView for that folder.
6. Enter the web address corresponding to this folder and server in the edit field.
  - a. Usually you should enter your domain name, or the address for your website's homepage.
  - b. Examples of web addresses are: `http://www.example.com/myusername/` or `http://www.MyDomainName.com`
  - c. You can use the **Get From Safari** pop-up menu to get the web address of a webpage currently open in Safari.
7. Click the **OK** button. Fetch should show the selected file or folder in your web browser.

If the webpage that opens after you have entered a web address is incorrect — that is, it doesn't show you the page you expected, or it shows you an error — you may have entered an incorrect web address and need to try again. To edit a WebView entry, either hold down the Option key and click the **WebView** button, or use the [WebView Preferences](#) pane.

Once you have set up WebView for a server, you do not have to go through the set up process again.

The next time you choose **WebView** when connected to the same server, Fetch will show the selected files or folder in your web browser instead of the WebView dialog.

If you work with multiple servers, you can set up WebView entries for each server.

Why doesn't Fetch automatically figure out the web address for you? Unfortunately, there's no standard for specifying the web address of files on a server; each server uses its own method, and the method used by one server may not apply on another. There is no way for Fetch to ask your server which method it uses, so Fetch has to ask you instead. While the information for some popular servers is built into Fetch, in most cases Fetch needs your input. If you are not sure what the corresponding web address should be, you should contact your hosting provider or server administrator.

Related topics:

- [WebView](#)
- [WebView Preferences](#)
- [WebView dialog \(web address only\)](#)
- [WebView command](#)



Using the suggestions below, you can solve some of the most common problems you may have uploading your website; if they don't help, contact Fetch support for further assistance.

1. [I don't know or I've forgotten my hostname, username, or password.](#)
2. [I don't see the latest changes I made to my website.](#)
3. [My website doesn't show up at all.](#)
4. [I get a "permission denied" or "operation not permitted" error when I try to upload my website files.](#)
5. [Some of my website's links do not work or some of my pictures do not appear.](#)

*I don't know or I've forgotten my hostname, username, or password.*

You should contact the company or person that runs the server to which you are trying to connect, and verify the hostname, username, and password you should use. When you are trying to upload webpages, usually this means you should contact the web hosting provider that you have signed up with or your Internet service provider. See the [What do I enter in the hostname, username, and password fields?](#) help topic for more information.

*I don't see the latest changes I made to my website.*

First, go to one of the pages you changed, and choose your web browser's reload or refresh command. In Safari, go to the **View** menu and choose **Reload Page**.

If you still don't see your changes, next empty your browser's cache. In Safari, go to the **Safari** menu and choose **Empty Cache**. Then quit your web browser, open it again, and visit your website and see if your changes appear now.

If you still don't see your updates after doing that, you may have uploaded your new webpages to a different folder than the one where the old webpages are. See the advice below for [My website doesn't show up at all](#).

*My website doesn't show up at all.*

Check to make sure you uploaded your website files to the correct folder on the server. On many servers, the files that make up your website must go in a specially named folder, for example, one named "public\_html", "htdocs", or "www" — the exact name of the folder will depend on your server. Usually you do not need to create this folder, it will already exist.

If you are unsure which folder you should put your website files in, you should contact the company or person that runs the server and ask which folder you should upload website files to.

If you used iWeb to create your website, be sure that you uploaded the contents of the folder where you published your website, but not the folder itself. (This advice does not necessarily apply if you are using a program other than iWeb to create your website.)

For example, let's assume your website is called "Family Trip", and when you use iWeb's **Publish to a Folder** command, you tell iWeb to put its files in a folder called "My New Website." Once iWeb is done publishing your files, in the Finder

you will see:

My New Website (folder)  
index.html  
Family Trip (folder)

For your website to work correctly, you must be sure to upload both that index file, and the "Family Trip" folder (the folder with the name of your website). That is, upload the entire contents of the "My New Website" folder, but don't upload the "My New Website" folder itself.

It may be best to delete the files that you've already uploaded to the server and start fresh before trying again.

*I get a "permission denied" or "operation not permitted" error when I try to upload my website files.*

There are two common causes for these errors.

First, you may be trying to upload your files to the wrong folder on the server, and you do not have permission to upload files to that folder because it's not the proper location to put files. Double-check the instructions from your web hosting provider to make sure you're uploading to the correct folder for website files. See the advice above for [My website doesn't show up at all](#).

Second, if you are using iWeb, the problem may be that some of the files and folders created by iWeb have spaces in their names, and some web servers do not allow spaces in the names of files or folders. Since iWeb uses the titles of your webpages as the names for files and folders, the solution is to edit your website so that none of the titles of your webpages contain spaces, and then re-publish the website and try uploading again. Do not rename your files in the Finder; this will break all your links. You must rename the pages in iWeb, and use iWeb's **Publish to Folder** command again.

If you rename your pages, it may be best to delete the files that you've already uploaded to the server and start fresh before trying again.

*Some of my website's links do not work or some of my pictures do not appear.*

If you rearranged or renamed files before uploading your website, that may be the source of the problem. Generally it is best to rename or rearrange websites using your website design program; do not rename or rearrange files in the Finder.

If you are using iWeb, after you make changes to your website, you must re-publish and re-upload the entire site. Because of the way iWeb creates websites, if you only upload some of your files after updating them, your website may not work properly. If you use the Fetch [Mirror](#) command to upload your website, it will correctly upload all the updated files as necessary.

Also, see the advice above for [I don't see the latest changes I made to my website](#).

See the following help topics for more information about using Fetch for uploading websites:

- [Tutorial: Uploading](#)
- [Using Fetch with iWeb](#)

## [Fetch Help](#) > [Using Fetch](#) > **Creating a shortcut**

Shortcuts are a quick way to access remote files or folders, similar to bookmarks in web browsers. They let you access items without typing in connection information each time.

After you've created a shortcut, you can access it from the Shortcuts  pop-up menu in the [New Connection dialog](#) or open it from the [Fetch Shortcuts](#) window.

There are several ways to create a shortcut.

To create a shortcut from the New Connection dialog:

1. Choose **File > New Connection**, if the New Connection dialog isn't already showing.
2. Enter the name of the server in the Hostname field. If your connection requires a username and password, enter them in the appropriate fields.
3. Click the Shortcuts  pop-up menu, and choose **Make Shortcut**.
4. Enter a name for the shortcut. This can be anything that makes it easy for you to identify the shortcut later on.
5. To make the information from the new shortcut always appear when you open the New Connection dialog, check the **Make this the default shortcut** checkbox.
6. Click the **OK** button.
7. Click **Connect** to connect and create the shortcut.

If you're not sure what information to enter in the New Connection dialog, see the [What do I enter in the hostname, username, and password fields?](#) help topic.

To create a shortcut to a file or folder in a transfer window:

1. Open a connection to the server where the desired file or folder is stored.
2. Select the item you want to create a shortcut to in the transfer window.
3. Choose **Shortcuts > New Shortcut**.
4. Enter a name for the shortcut. This can be anything that makes it easy for you to identify the shortcut later on. The rest of the information is already filled out for you.
5. Click **OK** to create the shortcut.

To create a shortcut by entering the location information manually when not in the New Connection dialog:

1. Make the Fetch Shortcuts window active (choose **Shortcuts > Show Shortcuts**).
2. Choose **Shortcuts > New Shortcut**.
3. Enter information in at least the Name and Hostname fields, and choose the desired connection type from the **Connect using** menu.
4. Optionally enter a username, password, and path. For more information about these choices, see the [New Shortcut dialog](#) help topic.
5. Click **OK** to create the shortcut.

You can also create shortcuts by:

- Dragging a file or folder from a transfer window to the Fetch Shortcuts window;
- Pasting a Fetch address (URL) from another application into the Fetch Shortcuts window.

See the [shortcuts](#) help topic for more information about shortcuts.

Related topics:

- [Deleting or changing a shortcut](#)
- [Droplet shortcuts](#)



## [Fetch Help](#) > [Using Fetch](#) > Deleting or changing a shortcut

You can delete a shortcut altogether, or update an existing shortcut without recreating it from scratch.

To delete a shortcut:

1. Choose **Shortcuts** > **Show Shortcuts**
2. Select the shortcut in the Fetch Shortcuts window.
3. Choose **Edit** > **Delete** , or press the Delete key.
4. A confirmation dialog will be displayed.

You can also delete a shortcut by dragging it from the Fetch Shortcuts window the Mac OS Trash.

To change an existing shortcut:

1. Choose **Shortcuts** > **Show Shortcuts**
2. Select the shortcut in the Fetch Shortcuts window.
3. Choose **Shortcuts** > **Edit Shortcut** .
4. Edit any of the fields in the Edit Shortcut dialog. For more information about these choices, see the [Edit Shortcut dialog](#) help topic.
5. Click **OK** to save your changes.

You can also open the Edit Shortcut dialog by Option–Shift–double–clicking a shortcut.

See the [shortcuts](#) help topic for more information about shortcuts.



## [Fetch Help](#) > [Using Fetch](#) > Moving shortcuts to another Macintosh

When you are upgrading to a new Macintosh, or if you use multiple Macintosh computers, you may want the same Fetch shortcuts on each computer.

To accomplish this, you need to follow a different set of instructions depending on which version of Fetch you were using on the old Macintosh. If you're not sure which version of Fetch you were using, open Fetch on the old Macintosh, choose **Fetch** > [About Fetch](#), and the version number will be displayed in the About window.

These instructions assume both computers are using Mac OS X.

If you were using Fetch 5.1 or later on the old Macintosh:

On the old Macintosh:

1. Open Fetch.
2. Choose **Shortcuts** > **Show Shortcuts**.
3. Choose **File** > **Save a Copy As** to export the shortcuts.
4. Pick a name to give the exported shortcuts file, such as "Old Shortcuts", and pick a location to save them.
5. Quit Fetch.
6. Copy the "Old Shortcuts" file to the new Macintosh.

On the new Macintosh:

1. Open Fetch.
2. Cancel the New Connection dialog if it appears.
3. Choose **Shortcuts** > **Show Shortcuts**.
4. In the Finder, double-click the "Old Shortcuts" file from your old computer.
5. You should now have two shortcut list windows open, one named "Fetch Shortcuts" and one named "Old Shortcuts."
6. Choose **Edit** > **Select All** in the window containing your old shortcuts.
7. Drag the selected shortcuts to the "Fetch Shortcuts" window.
8. Close the window containing your old shortcuts.

If you were using Fetch 5.0.5 or earlier on the old Macintosh:

1. Locate the "Fetch Shortcuts" file on your old Macintosh. One way to do this is to choose **Go** > **Home** in the Finder of the old Macintosh, double-click on the Library folder, and then double-click on the Preferences folder. The Fetch Shortcuts file is in that folder.
2. Copy the file to the new Macintosh.
3. Open Fetch on the new Macintosh.
4. Cancel the New Connection dialog if it appears.
5. Choose **Shortcuts** > **Show Shortcuts**.
6. In the Finder, double-click the "Fetch Shortcuts" file from your old computer.
7. You should now have two shortcut list windows open, both named "Fetch Shortcuts".
8. Choose **Edit** > **Select All** in the window containing your old shortcuts.
9. Drag the selected shortcuts to the other "Fetch Shortcuts" window.

## 10. Close the window containing your old shortcuts.

Since Fetch 5.1 and later store passwords for shortcuts in your [keychain](#), moving the shortcuts file from one Macintosh with Fetch 5.1 or later to another will move all the information about your shortcuts except the passwords.

The simplest solution, especially if you just have a few passwords, is to re-enter your passwords on the new Macintosh (where they will be saved in that Macintosh's keychain). Otherwise, see the [moving keychains and passwords to another Macintosh](#) help topic for more options.

If you are upgrading from a previous version of Fetch on the same computer, you do not need to import your old shortcuts. Fetch will do that automatically.

See the [shortcuts](#) help topic for more information about shortcuts.

Related topics:

- [Backing up your shortcuts](#)



## [Fetch Help](#) > [Using Fetch](#) > Backing up your shortcuts

Fetch does not automatically make a backup copy of your shortcuts. If you want to back up your [shortcuts](#) — as displayed by the Shortcuts  pop-up menu or in the [Fetch Shortcuts](#) window — to a file for safe-keeping, use the [Save a Copy As](#) menu command:

1. Choose **Shortcuts > Show Shortcuts** to display the Fetch Shortcuts window.
2. Choose **File > Save a Copy As**.
3. You will be prompted to name the file and choose a location to save it.
4. Click **Save**, and a new file containing all your shortcuts will be saved to your disk.

Note that this only saves your shortcuts; it will not save your [recent connections](#). There is no way inside Fetch to back up your recent connections, although if you back up your Macintosh they will be backed up as part of the Fetch preferences (com.fetchsoftworks.Fetch.plist). You should [create a shortcut](#) for any important recent connection.

To restore your shortcuts after you have used the steps above:

1. Either:
  - In the Finder, double-click the backup file you created using the steps above;
  - Or choose **File > Open** in Fetch, and navigate to and select the backup file you created using the steps above.
2. You should now have two shortcut list windows open, one or both named "Fetch Shortcuts".
3. Choose **Edit > Select All** in the window containing the shortcuts you want to restore.
4. Drag the selected shortcuts to the other "Fetch Shortcuts" window.
5. Close the window containing your backed-up shortcuts.

See the [shortcuts](#) help topic for more information about shortcuts.

Related topics:

- [Moving shortcuts to another Macintosh](#)
- [Creating a shortcut](#)
- [Deleting or changing a shortcut](#)



## [Fetch Help](#) > [Using Fetch](#) > **Mirroring a local folder to a remote folder**

The Mirror window lets you automatically copy all new or changed files from a local Macintosh folder to a remote folder on a server. For example, if you keep all the files that make up a website in a folder on your Macintosh, mirroring can quickly upload only the files that are new or have changed since your last update, instead of using the [Put](#) command to upload the entire site again.

To mirror a local folder to a remote folder:

1. Choose **Remote > Mirror** .
2. Make sure the **Local** tab is selected in the left-hand pane of the Mirror window.
3. Click the **Choose** button in the left-hand pane.
4. Select a local folder from the "Choose a folder" prompt.
5. In the right-hand **Remote** pane, click the **Choose** button.
6. A dialog similar to the New Connection dialog appears. Either enter the information for the remote folder, or choose one of your shortcuts or a recent connection from the pop-up menus.
7. Select the top arrow button (pointing from left to right) between the two panes to indicate the direction of mirroring.
8. Click **Mirror** to open the connection and start mirroring.

Note that the first time you mirror a local folder to a remote folder, Fetch will most likely upload all your files, because it needs to start by synchronizing all the files. But the next and all subsequent times you use the Mirror window for the same folders, only new or changed files will get uploaded.

If you're going to mirror these folders often, you can save these mirror settings as a [mirror document](#). This will make uploading the changes a simple one click process:

1. With the information from above still entered in the Mirror window, click the **Save** button.
2. Choose a location and name for the mirror document.
3. Check the **Mirror automatically when opened** box to tell Fetch to automatically start mirroring when when you open the mirror document. Otherwise, Fetch will open the Mirror window with the settings saved in the document, and you will have to click the **Mirror** button to start mirroring.
4. Click the **Save** button.

From now on, you can just open the mirror document instead of selecting the local and remote folders each time.

You can also mirror a remote folder to a local folder (choose the bottom arrow button instead of the top), or mirror two remote folders (click the Remote tab in the left-hand pane, and choose two remote folders).

For more information about mirroring and the Mirror window, see the [introductory](#) or [in-depth](#) Mirror window help topics.



## [Fetch Help](#) > [Using Fetch](#) > Editing a file

Fetch's **Edit** command lets you make changes to any kind of file in another application and save the changes back to the server automatically.

To edit a file:

1. Connect to the server where the file is located.
2. Select the file that you want to edit by clicking it.
3. Click the **Edit** button in the toolbar, or choose **Remote** > **Edit** .
4. The editor application for that kind of file opens. Make changes to the file.
5. In the editor, choose **File** > **Save** . Fetch will automatically upload the changed file back to the server.

If Fetch does not open the application you would like to use for editing a certain kind of file — or you would like to use a different editor temporarily — hold down the Option key when clicking the **Edit** button or choosing the **Edit** menu item. Fetch will ask you to choose a different application to use for editing and whether you want Fetch to remember this choice for the future or not.

You can also choose a different editor to use temporarily with the [Edit With submenu](#) in the **Remote** menu or the contextual menus, or see the [changing the editor for a file](#) help topic for another method of specifying editors.

For more information, see the [Edit](#) command help topic.

Related topics:

- [Editing files automatically instead of downloading](#)
- [Edit With](#)
- [Edit New Text File](#)
- [Edit Other](#)



## [Fetch Help](#) > [Using Fetch](#) > Changing the editor for a file

When you use the **Edit** command to make changes to a file, Fetch decides which editor application to use based on the file's extension — the part of the filename at the end after a period, such as ".txt" or ".jpg" . Usually the default editor for a kind of file is the same as the application that would normally open that kind of file, although in certain cases the editor may be different (for instance, when the usual application cannot make edits to files).

To change the editor for a kind of file, use the Transfer Options pane of the [info window](#):

1. Connect to your server, and find a file with the extension whose editor you want to change. You may need to upload an example file if you do not know where one is located.
2. Select the file.
3. Click the **Get Info** button, or choose **Remote > Get Info** .
4. Click the disclosure triangle next to the Transfer Options label at the bottom of the info window to display the Transfer Options pane, if it's not already displayed.
5. Click the pop-up menu labeled **Edit files like this with**.
6. Choose an application from the menu, or if the desired application is not listed, choose the **Other** item and select the desired application.

From now on, any time you edit a file with the same extension as the one whose transfer options you just changed, the newly selected editor will be used.

Note that changing the editor only affects which application will be used by Fetch's **Edit** command. It does not change which application will be used to open a file after you download it; for information on how to do that, see the [changing the application a file opens with](#) help topic.

You can also change the editor for a file by holding down the Option key when choosing the **Edit** menu item or clicking the **Edit** button, then choosing a new application and checking the **Remember this editor choice** checkbox.

The Transfer Options pane of the info window can also be used to tell Fetch to [automatically edit files](#) instead of downloading them or to [open certain kinds of files automatically](#) after they've been downloaded. See the [Transfer Options](#) help topic for more information.

Related topics:

- [Get Info](#)
- [Info window](#)
- [Download Preferences](#)



## [Fetch Help](#) > [Using Fetch](#) > **Editing files automatically instead of downloading**

You can tell Fetch to automatically edit files instead of downloading them when you double-click them by using the [Transfer Options](#) pane of the [info window](#).

When you edit a file with Fetch, the file is temporarily downloaded to your Macintosh, opened in the editor application, and then saved back to the server when you use the editor application's **Save** command. A copy of the file will not be left on your Macintosh. You can any kind of file with any application on your Macintosh. See the [Edit](#) command help topic for more information about editing files.

To set a certain kind of file to be edited automatically:

1. Connect to your server, and find a file whose name ends with an extension for the kind of file that you want to automatically edit. For example, for text files, find a file that ends in ".txt".
2. Select the file.
3. Choose **Remote > Get Info** .
4. Click the disclosure triangle next to the Transfer Options label at the bottom of the info window to display the Transfer Options pane, if it's not already displayed.
5. In the "After transferring a file like this" section, click the **Edit the file** button.

From now on, any time you double-click on a file with the same extension it will automatically be opened for editing in the application listed in the **Edit files like this with** pop-up menu in the Transfer Options pane. Choose a different application from that menu to change the editor application for this kind of file.

Even if the **Edit the file** option is set, a file will not be edited when you click the **Get** button or drag the file to the Mac OS Finder — this option only applies when you double-click a file.

You can set as many different kinds of files as you like to be automatically edited.

To stop automatically editing a kind of file, follow the instructions above, but click the **Save the file** button in the "After transferring a file like this" section.

You can also use the Transfer Options pane of the info window to [automatically open certain kinds of files after downloading](#). Also see the [Transfer Options](#) help topic for more information.

Related topics:

- [Get Info](#)
- [Info window](#)



## [Fetch Help](#) > [Using Fetch](#) > **Fixing problems choosing editors and helper applications**

Fetch allows you to choose other applications as helpers for editing or opening files, or to choose an application to be the default FTP or SFTP helper. Problems choosing helper applications are the result of a problem in Mac OS X that prevents Fetch from properly detecting which applications you have installed, and which files belong to which applications. These problems include:

- Fetch does not list an application for opening or editing files in the **Open files like this with** or **Edit files like this with** pop-up menus in the info window, even though the application is installed
- Fetch does not remember your choice for the application to edit or open a kind of file after you choose it
- Fetch (or another FTP application) is not listed as a choice for the default FTP or SFTP application in the [General Preferences](#) pane, or Fetch does not remember your choice after you choose it

To fix any of these problems, you should first try the following:

1. Quit Fetch.
2. In the Finder, empty the trash (choose **Finder** > **Empty Trash**).
3. Open the application you would like to use as a helper application.
4. Open Fetch again and see if the problem has gone away; that is, see if the desired application is listed or if your application choice is remembered correctly.

If this does not fix the problem, your Launch Services database needs to be rebuilt. Launch Services is the part of Mac OS X that keeps track of which applications can open which kinds of files—unfortunately, sometimes it can get confused and lose track of some applications. The link below will run a script to rebuild your Launch Services database.

Clicking on the link below will quit Fetch and will empty your trash; be sure you want to do that before clicking on the link.

[Rebuild my Launch Services database for me](#)

**IMPORTANT:** After running the script, you must open the helper applications you want Fetch to use before opening Fetch again; otherwise, Fetch may not list them. After opening the helper applications and then opening Fetch, Fetch should list the applications and remember your choices correctly.

After you run this script, you may be warned that you are opening applications for the first time, even though it is not the first time you have opened them. This is normal.



You can use Fetch's [transfer window](#) to move files and folders from one folder to another folder on your server.

To move files or folders to a subfolder of the current folder, select the items and drag them to the subfolder's icon. For example, if you have FolderOne and FolderTwo, drag FolderOne onto FolderTwo to move it into FolderTwo.

If your list of files is long enough that you can't see the files you want to move and the subfolder you want to put them in at the same time, the transfer window will scroll as you drag the items up or down.

To move items up a level, that is, into the parent folder of the current folder, drag them to the **Parent** button in the toolbar. Fetch 5 does not display the **Parent** button by default, you must first use **View > Customize Toolbar** to add it to the toolbar if it is not already displayed.

To move items to a different folder that is not a subfolder or a parent folder of the current folder, you should open another transfer window. To do this:

1. Control-click in the transfer window and choose [Clone Window](#) from the contextual pop-up menu. This will create a second transfer window that is connected to the same server and folder as the original transfer window.
2. Navigate to the folder you want to move files to in the new window.
3. Drag items from the old window to the new window to move them.

You can also open a folder into a new transfer window by Command-double-clicking on the folder.

If you want to copy items instead of moving them, hold down the Option key while dragging them. The cursor will change to an arrow with a plus symbol , indicating that the files will be copied instead of moved.

You can copy items between two servers by opening connections to both servers, and then dragging the items from one transfer window to the other.

Another way to copy files and folders from one folder to another on the same server, or from one server to another, is to use the **Copy** and **Paste** commands. See the [copying files using copy and paste](#) help topic for more information.



## [Fetch Help](#) > [Using Fetch](#) > Copying files using copy and paste

You can use the standard Macintosh Copy and Paste commands to copy files and folders from one folder to another on the same server, or from one server to another.

To do this:

1. In a Fetch transfer window, select the files and folders you want to copy.
2. Choose **Edit > Copy Fetch Address** (or **Edit > Copy Web Address** if you've configured WebView for this server.)
3. Navigate to a new folder on the same server, or open a connection to a new server.
4. Choose **Edit > Paste**.

The files will be copied from the original folder to the new folder using a server-to-server transfer. If you are copying files from one folder to another on the same server, Fetch may open a new transfer window to accomplish the task. The original copies of the files will remain in the original folder.

Related topics:

- [Uploading files using copy and paste](#)
- [Server-to-server transfers](#)



## [Fetch Help](#) > [Using Fetch](#) > Renaming files and folders

You can rename a file or folder by clicking in the item's name in the [file list](#), or using the "Name and extension" field in the [info window](#).

Renaming a file or folder in the file list works like renaming an item in the Finder:

1. Select the item you want to rename by clicking it.
2. Press the Return key to display the edit box.
3. Type in a new name or edit the existing name.
4. Press Return again when you're done.

To rename a file or folder using the info window:

1. Select the item you want to rename.
2. Choose **Remote** > **Get Info** or click the **Get Info** button in the transfer window toolbar.
3. Click the disclosure triangle next to "Name and Extension" to display the name field, if necessary.
4. Enter the new name and extension for the file.
5. Click the **Apply** button.

When renaming items, try to choose a name that will be acceptable to the remote computer: some systems have filename length limits and do not allow certain characters. If you specify an improper name, you will get an error message.

If you do not have permission to make changes to the item, you will receive an error when you try to rename it.

On UNIX servers, and some others, it's possible to move a file to a different folder by renaming it. For example, renaming "myfile" to "other/myfile" moves it into the "other" folder.

Related topics:

- [Get Info](#)
- [File list](#)

## [Fetch Help](#) > [Using Fetch](#) > **Changing folders**

There are many different ways you can change to a different remote folder in Fetch. (You can tell which remote folder you are currently in by looking at the title bar of the transfer window, where the name of the current folder is displayed. If you are in your home folder, you'll see a house icon.)

In the file list, you can double-click a remote folder to make that subfolder the current folder. You can also select a folder and press Command-O or Command-Down arrow to make it the current folder.

To return to a folder you were previously viewing, click the [Back](#) button in the transfer window toolbar, or choose **Go > Back**.

Folders you have recently visited are listed at the bottom of the **Go** menu, or when you click the [Recent](#) button in the transfer window toolbar. Choose one of the folders listed to make that folder the current folder.

If you click the [Path](#) button in the transfer window toolbar, a pop-up menu shows a list of all the folders containing the current folder. Choose an item from the menu to make that the current folder. You can also get a path pop-up menu by Command-clicking the transfer window title bar.

You can go to the parent folder (the folder enclosing the current folder) by choosing **Go > Parent Folder** or pressing Command-Up arrow.

If you know the path of a folder you want to go to, you can go there directly by choosing **Go > Go to Folder**, and typing in the path.

Finally, you can return to your home folder — the folder you are taken to when you first connect to a server — by choosing **Go > Home**.

You can add **Parent Folder**, **Go to Folder**, and **Home** buttons to the transfer window toolbar with the **View > Customize Toolbar** command.

## [Fetch Help](#) > [Using Fetch](#) > Viewing and setting permissions

Permissions restrict which users can view, get, edit, put, or delete files, and which users can look inside or make changes to the contents of folders. For instance, you may have read permission (permission to look at a file on a server or download it) but not write permission (permission to delete a file or make changes to it).

You can view remote items' permissions with the [Get Info](#) command, and change their permissions (if you are the owner of the items). Typically you can only view or set permissions on servers running the UNIX operating system.

You can set read, write, and execute permissions for three sets of users — the owner, the group, and all others (world permission).

To view the permissions of remote items:

1. Select the items you want to view the permissions of.
2. Choose **Remote** > **Get Info** or click the **Get Info** button in the transfer window toolbar.
3. Click the disclosure triangle next to "Ownership and Permissions" to display the permissions pane, if necessary.

If all the permissions checkboxes are blank, you are probably trying to view the permissions of an item you just uploaded. To see the true state of the permissions, close the info window, click the [Refresh](#)  button in the transfer window, and then get info on the item again.

To set or change the permissions of remote items:

1. Select the items you want to set permissions for.
2. Choose **Remote** > **Get Info** or click the **Get Info** button in the transfer window toolbar.
3. Click the disclosure triangle next to "Ownership and Permissions" to display the permissions pane, if necessary.
4. Check the permission boxes as desired, or enter a UNIX equivalent string in the edit field.
5. Click the **Apply** button.

To discard any changes to permissions, just close the info window without clicking the **Apply** button.

If you have selected multiple items, some of the checkboxes may have a dash instead of being unchecked or checked. That means not all of the items you've selected have the same permissions. Leaving a checkbox with a dash unchanged will preserve the mixed permissions, even if you change other permissions. If you click a checkbox with a dash, you can force the permissions to all become the same.

The permissions checkboxes are simultaneously translated into a UNIX equivalent string, which is shown in an edit field below the checkboxes. You can change the permissions by typing a UNIX equivalent string into that edit field.

See the [permissions](#) help topic for more information.



## 10.5 and 10.6

While Fetch does not have a built-in scheduler, you can use iCal along with Fetch's Mirror command or Automator support to automatically upload or download files at a specified time every day or week.

iCal is a calendar and scheduling application that is part of Mac OS X. In iCal, you create events on a calendar to remind yourself of meetings, appointments, birthdays, etc. You can create events that occur just once, or events that occur on a repeating schedule, such as every day, every week on a certain day, or every Monday through Friday. For each event, you can specify an alarm. While this can be an alarm in the traditional sense, where your Macintosh plays a sound or displays a message reminding you of the event, an iCal alarm can also open a file, send an email, or run an AppleScript. By creating a repeating event with an alarm that opens a file, you can use iCal to schedule automatic transfers in Fetch.

Once you've set up an event, iCal doesn't even need to be open for the event, and thus the transfer, to happen. As long as your Macintosh is on (and not sleeping), the transfer will happen at the time you specified.

This help topic discusses iCal on Mac OS X 10.5 Leopard and 10.6 Snow Leopard. If you are using Mac OS X 10.4 Tiger, you should see the [Mac OS X 10.4 help topic](#) instead.

### Setting up the transfer

First, you need to create a set of instructions telling Fetch what to do when the event's alarm goes off. Typically you want to do one of three things: mirror a folder, download certain files, or upload certain files.

If you want to transfer a whole folder, the easiest solution may be to create a [mirror document](#) with Fetch's [Mirror](#) command. The Mirror command transfers the contents of a local folder to a server folder. One benefit of the Mirror command is that it will only transfer the files that are new or have changed since the last time you mirrored the folder, saving you time. For more information and step-by-step instructions on creating a mirror document, see the [mirroring a local folder to a remote folder](#) help topic. When creating a mirror document for scheduled transfers, be sure to check the **Mirror automatically when opened** checkbox.

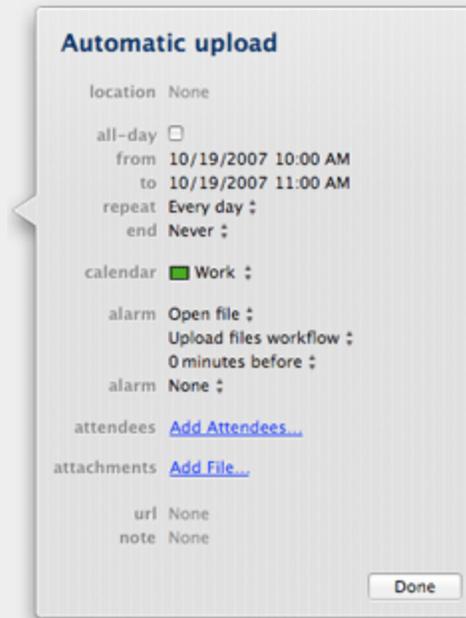
Downloading and uploading certain individual files is best accomplished by creating an Automator workflow that uses Fetch's Automator actions. For an overview of using Automator and instructions on creating simple workflows for uploading and downloading files, see the [introduction to using Automator with Fetch](#) help topic.

### Scheduling the automatic transfer in iCal

Next, you need to create the event in iCal and attach your mirror document or Automator workflow to it.

1. Open iCal (located in your Applications folder).
2. Select the date you want the automatic transfer to start happening.
3. Choose **File > New Event**. A new event will appear in the iCal window.
4. Choose **Edit > Edit Event**, and the event editor will appear. When you are

done following the steps below, the event editor should look something like this:



5. Click the name at the top of the event editor, and enter a new name for the event so you can easily identify it later.
6. On the **from** line, enter a start time by clicking the hour and minutes and typing in new times.
7. Skip the **to** line, which specifies the end time. The end time isn't important when setting up an event to do automatic transfers.
8. Skip the repeat line for now, but we'll come back to it.
9. Next, click the triangles in the **alarm** line, and choose **Open file** from the pop-up menu. Two new lines will appear beneath the **alarm** line.
10. Click the first line beneath the alarm line (most likely it says "iCal"), and choose **Other** from the pop-up menu.
11. iCal will prompt you to select a file. Find and select the mirror document or Automator workflow that you created earlier.
12. Then, in the next line, which typically says "15 minutes before," click the number and change it to "0", so that your alarm happens at exactly the start time specified for this event.
13. A new, empty **alarm** line will appear once you have finished creating the first alarm, in case you want to set two different alarms for the same event. Usually you do not.
14. Finally, return to the **repeat** line and click the triangles to specify how often you want the event and alarm to repeat. From the pop-up menu, choose **every day**, **every week**, or **Custom** (if you want the event to happen on some days of the week, but not other days).
15. Once you've chosen a repeat schedule, the **end** line will appear. Usually you want to leave this as "Never," but if, for example, you want your automatic transfers to only happen during the next month, you can specify a date for the automatic transfers to end.

Your automatic transfer is now set up. If you want, you can view other days in the calendar to verify that the event has been added to all the dates specified.

If you want to set up more than one scheduled automatic transfer, repeat the steps above.

You can quit iCal, since events and alarms will trigger even when iCal isn't running.

When you want to turn off a scheduled transfer, open iCal, click the event for the transfer, and press the Delete key to remove it and all future occurrences of it.



## 10.4

While Fetch does not have a built-in scheduler, you can use iCal along with Fetch's Mirror command or Automator support to automatically upload or download files at a specified time every day or week.

iCal is a calendar and scheduling application that is part of Mac OS X. In iCal, you create events on a calendar to remind yourself of meetings, appointments, birthdays, etc. You can create events that occur just once, or events that occur on a repeating schedule, such as every day, every week on a certain day, or every Monday through Friday. For each event, you can specify an alarm. While this can be an alarm in the traditional sense, where your Macintosh plays a sound or displays a message reminding you of the event, an iCal alarm can also open a file, send an email, or run an AppleScript. By creating a repeating event with an alarm that opens a file, you can use iCal to schedule automatic transfers in Fetch.

Once you've set up an event, iCal doesn't even need to be open for the event, and thus the transfer, to happen. As long as your Macintosh is on (and not sleeping), the transfer will happen at the time you specified.

This help topic discusses iCal on Mac OS X 10.4 Tiger. If you are using Mac OS X 10.5 Leopard or 10.6 Snow Leopard, you should see the [Mac OS X 10.5 and 10.6 help topic](#) instead.

### Setting up the transfer

First, you need to create a set of instructions telling Fetch what to do when the event's alarm goes off. Typically you want to do one of three things: mirror a folder, download certain files, or upload certain files.

If you want to transfer a whole folder, the easiest solution may be to create a [mirror document](#) with Fetch's [Mirror](#) command. The Mirror command transfers the contents of a local folder to a server folder. One benefit of the Mirror command is that it will only transfer the files that are new or have changed since the last time you mirrored the folder, saving you time. For more information and step-by-step instructions on creating a mirror document, see the [mirroring a local folder to a remote folder](#) help topic. When creating a mirror document for scheduled transfers, be sure to check the **Mirror automatically when opened** checkbox.

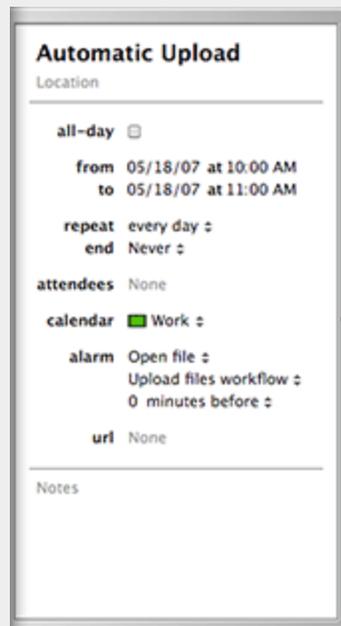
Downloading and uploading certain individual files is best accomplished by creating an Automator workflow that uses Fetch's Automator actions. For an overview of using Automator and instructions on creating simple workflows for uploading and downloading files, see the [introduction to using Automator with Fetch](#) help topic.

### Scheduling the automatic transfer in iCal

Next, you need to create the event in iCal and attach your mirror document or Automator workflow to it.

1. Open iCal (located in your Applications folder).
2. Select the date you want the automatic transfer to start happening.
3. Choose **File > New Event**. A new event will appear in the iCal window, and the Info drawer for that event should appear on the right-hand side of the

window. If you do not see the Info drawer, choose **View > Show Info**. When you are done following these instructions, the Info drawer should look something like this:



4. Click the name at the top of the Info drawer, and enter a new name for the event so you can easily identify it later.
5. On the **from** line, enter a start time by clicking the hour and minutes and typing in new times.
6. Skip the **to** line, which specifies the end time. The end time isn't important when setting up an event to do automatic transfers.
7. Skip the repeat line for now, but we'll come back to it.
8. Next, click the triangles in the **alarm** line, and choose **Open file** from the pop-up menu. Two new lines will appear beneath the **alarm** line.
9. Click the first line beneath the alarm line (most likely it says "iCal"), and choose **Other** from the pop-up menu.
10. iCal will prompt you to select a file. Find and select the mirror document or Automator workflow that you created earlier.
11. Then, in the next line, which typically says "15 minutes before," click the number and change it to "0", so that your alarm happens at exactly the start time specified for this event.
12. Finally, return to the **repeat** line and click the triangles to specify how often you want the event and alarm to repeat. From the pop-up menu, choose **every day**, **every week**, or **Custom** (if you want the event to happen on some days of the week, but not other days).
13. Once you've chosen a repeat schedule, the **end** line will appear. Usually you want to leave this as "Never," but if, for example, you want your automatic transfers to only happen during the next month, you can specify a date for the automatic transfers to end.

Your automatic transfer is now set up. If you want, you can view other days in the calendar to verify that the event has been added to all the dates specified.

If you want to set up more than one scheduled automatic transfer, repeat the steps above.

You can quit iCal, since events and alarms will trigger even when iCal isn't running.

When you want to turn off a scheduled transfer, open iCal, click the event for the transfer, and press the Delete key to remove it and all future occurrences of it.

## [Fetch Help](#) > [Using Fetch](#) > Introduction to using Automator with Fetch on Mac OS X 10.5 and 10.6

Fetch includes a number of actions for Mac OS X's Automator, allowing you to create simple workflows that can automate a variety of Fetch tasks without having to know programming.

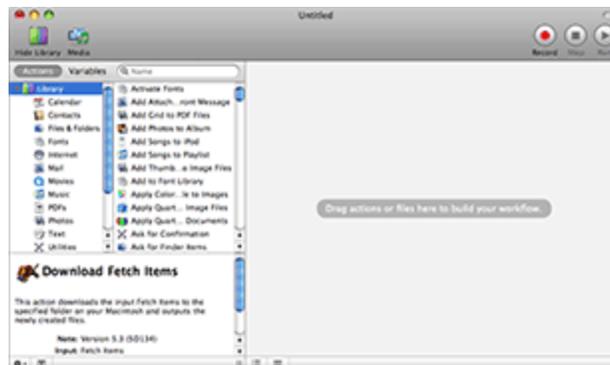
This help topic includes a brief overview of Automator, and step-by-step instructions for creating two simple workflows that upload and download specific files using Fetch. While only three Fetch actions are mentioned here, Fetch includes a variety of [other Automator actions](#) as well.

This help topic discusses Automator on Mac OS X 10.5 Leopard and 10.6 Snow Leopard. If you're using Mac OS X 10.4 Tiger, you should see the [Mac OS X 10.4 help topic](#) instead.

### Automator overview

Automator lets you put together individual steps, called actions, into a complete task, called a workflow. You do not have to be a programmer to use Automator — Automator lets you assemble actions into workflows in an easy-to-use graphical interface. You can then save these workflows and run them to perform the same task over and over again, without having to set it up each time.

When you open Automator, it will ask you to select a template or starting point for a new workflow. For creating workflows that automate Fetch, you should select the default template ("Custom" or "Workflow" depending on your version of Mac OS X), and click the **Choose** button. Then Automator will show a new, blank workflow window:



There are two lists on the left side of the window — the Library list, showing categories of similar Automator actions, and the Action list, showing the actions that belong to the selected category. To view the actions that belong to a certain category, click the category's icon in the Library list. The Fetch actions are in the Internet category.

To create an Automator workflow, you drag actions from the Action list to the big workflow area on the right side of the window. Most actions require additional information — they have buttons, fields, and pop-up menus that you should use to tailor the action to the needs of your task.

To see what each action does, select the action in the Action list, and read the action description in the lower-left corner of the window. You can also see an action's description by clicking the Description label at the bottom of the action.

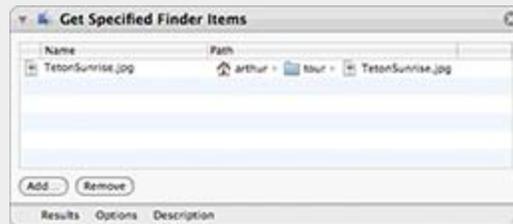
As you add actions, they will join up to form the workflow — you will see triangles leading from one action to the next. To remove an action from your workflow, click the Delete (x) icon in the action's title bar.

This is only a brief summary about Automator, which should give you enough information to get started. The Automator Help book, available in the Automator application by choosing **Help > Automator Help**, provides much more information and has an excellent walk-through of creating Automator workflows. You may also want to read the [Mac OS X help topics on Automator](#).

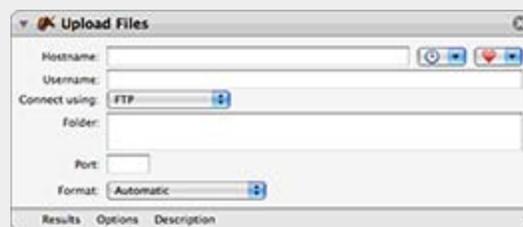
## Uploading files workflow

Follow these steps to create an Automator workflow that uploads specific files or folders from your Macintosh to a server. This can be useful if you frequently upload the same set of files, or if you want to set up [scheduled uploads using iCal](#).

1. Open Automator (located in your Applications folder). When prompted, click the **Choose** button to create a new "custom" workflow. A new workflow window should open automatically. If it does not, choose **File > New**.
2. In the new workflow window, click the Files & Folders icon in the Library list.
3. Drag a Get Specified Finder Items action from the Action list to the workflow area on the right side of the window.



4. In the Get Specified Finder Items action, click the Add button to add a file or folder to the list of items to be uploaded. When you click the Add button, a file selection prompt will appear. Choose the first file or folder you want to upload. Its name and path will appear in the Get Specified Finder Items' list.
5. Use the Add button to add as many files or folders as you want.
6. Next, click the Internet icon in the Library list.
7. Drag an Upload Files action from the Action list to the workflow area, placing it after the first action. (You may have to scroll the list to find the Upload Files action.)



8. In the Upload Files action, enter the connection information for the server and remote folder (if any) you want to upload the files to, or choose a recent connection or shortcut from the pop-up menus.

9. Finally, save the workflow as an application. Choose **File > Save As**. In the Save dialog, choose **Application** from the **File Format** pop-up menu, and then enter a name and choose a location to save your workflow.

Now, when you open the workflow, it will automatically open Fetch and upload the items you specified to the server you specified.

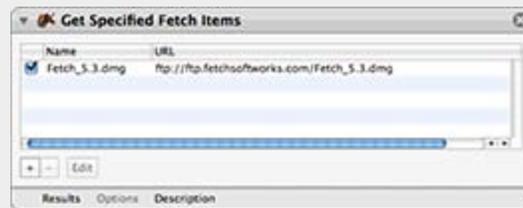
If the connection to the server requires a password, you will be prompted for it when you use the workflow. If you do not want to be prompted for the password, you should save the password for the connection in your [keychain](#).

If you want to edit your workflow later, drag the workflow's icon to the Automator application icon.

## Downloading files workflow

Follow these steps to create an Automator workflow that downloads specific files or folders from a server to your Macintosh. This can be useful if you frequently download the same set of files, or if you want to set up [scheduled downloads using iCal](#).

1. Open Automator (located in your Applications folder). When prompted, click the **Choose** button to create a new "custom" workflow. A new workflow window should open automatically. If it does not, choose **File > New**.
2. In the new workflow window, click the Internet icon in the Library list.
3. Drag a Get Specified Fetch Items action from the Action list to the workflow area on the right side of the window.



4. In the Get Specified Fetch Items action, click the Add (+) button to add a remote file or folder to the list of items to download. When you click the Add button, a dialog similar to Fetch's New Connection dialog will appear.
5. Enter the connection information for the server where the file is stored, and a path to the item. (A path is the description of a file or folder's location that includes the folders the item is in, for example, "www/pictures/dog.jpg" is the path to the file "dog.jpg" that is in the folder "pictures" that in turn is in the folder "www".)
6. Use the Add (+) button to add as many files or folders as you want.
7. Next, drag a Download Fetch Items action from the Action list to the workflow area, placing it after the first action.



8. In the Download Fetch Items action, click the pop-up menu to choose the folder on your Macintosh where you want the files to be saved. If you do not want to use any of the listed folders, choose **Other** to select any folder

on your Macintosh.

9. Finally, save the workflow as an application. Choose **File > Save As**. In the Save dialog, choose **Application** from the **File Format** pop-up menu, and then enter a name and choose a location to save your workflow.

Now, when you open the workflow, it will automatically open Fetch and download the items you specified to your Macintosh.

If the connection to the server requires a password, you will be prompted for it when you use the workflow. If you do not want to be prompted for the password, you should save the password for the connection in your [keychain](#).

If you want to edit your workflow later, drag the workflow's icon to the Automator application icon.

### More information and more examples

The two examples above are very simple examples of how you can use Automator with Fetch, using only a few of Fetch's Automator actions.

Other Fetch Automator actions let you rename, delete, or move files around on a server; set the permissions of items on a server; and filter items so that you download only items that meet a certain criterion and ignore other items. (There is also a Finder action for filtering Finder items, so that you upload only items that meet a certain criterion.)

For more information about each Fetch Automator action, see the [Automator actions](#) help topic.

For more examples of things you can do with Automator and Fetch using slightly more complex workflows, see the [more examples of using Automator With Fetch](#) help topic. It includes instructions of how to create workflows that batch change permissions of files on a server, download all files on a server that were created today, and more.



## 10.4

Fetch includes a number of actions for Mac OS X's Automator, allowing you to create simple workflows that can automate a variety of Fetch tasks without having to know programming.

This help topic includes a brief overview of Automator, and step-by-step instructions for creating two simple workflows that upload and download specific files using Fetch. While only three Fetch actions are mentioned here, Fetch includes a variety of [other Automator actions](#) as well.

This help topic discusses Automator on Mac OS X 10.4 Tiger. If you are using Mac OS X 10.5 Leopard or 10.6 Snow Leopard, you should see the [Mac OS X 10.5 and 10.6 topic](#) instead.

### Automator overview

Automator lets you put together individual steps, called actions, into a complete task, called a workflow. You do not have to be a programmer to use Automator — Automator lets you assemble actions into workflows in an easy-to-use graphical interface. You can then save these workflows and run them to perform the same task over and over again, without having to set it up each time.

When you open Automator, it will show a new, blank workflow window:



There are two lists on the left side of the window — the Library list, showing all applications that include Automator support, and the Action list, showing the actions for the selected application. To view an application's actions, click the application's icon in the Library list.

To create an Automator workflow, you drag actions from the Action list to the big workflow area on the right side of the window. Most actions require additional information — they have buttons, fields, and pop-up menus that you should use to tailor the action to the needs of your task.

To see what each action does, select the action in the Action list, and read the action description in the lower-left corner of the window. You can also see an action's description by clicking the action's title bar.

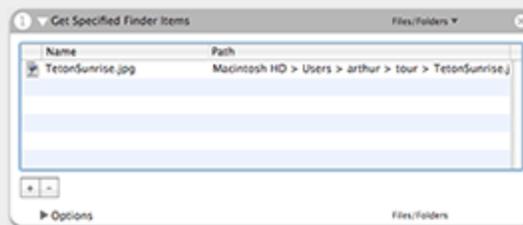
As you add actions, they will join up to form the workflow — you will see triangles leading from one action to the next. If you do not see a connection from one action to the next, that means the two actions cannot work together directly. To remove an action from your workflow, click the Delete (x) icon in the action's title bar.

This is only a brief summary about Automator, which should give you enough information to get started. The Automator Help book, available in the Automator application by choosing **Help > Automator Help**, provides much more information and has an excellent walk-through of creating Automator workflows. You may also want to read the [Mac OS X help topics on Automator](#).

## Uploading files workflow

Follow these steps to create an Automator workflow that uploads specific files or folders from your Macintosh to a server. This can be useful if you frequently upload the same set of files, or if you want to set up [scheduled uploads using iCal](#).

1. Open Automator (located in your Applications folder). A new workflow window should open automatically. If it does not, choose **File > New**.
2. In the new workflow window, click the Finder icon in the Applications list.
3. Drag a Get Specified Finder Items action from the Action list to the workflow area on the right side of the window.



4. In the Get Specified Finder Items action, click the Add (+) button to add a file or folder to the list of items to be uploaded. When you click the Add button, a file selection prompt will appear. Choose the first file or folder you want to upload. Its name and path will appear in the Get Specified Finder Items' list.
5. Use the Add (+) button to add as many files or folders as you want.
6. Next, click the Fetch icon in the Applications list.
7. Drag an Upload Files action from the Action list to the workflow area, placing it after the first action.



8. In the Upload Files action, enter the connection information for the server and remote folder (if any) you want to upload the files to, or choose a recent connection or shortcut from the pop-up menus.
9. Finally, save the workflow as an application. Choose **File > Save As**. In the Save dialog, choose **Application** from the **File Format** pop-up menu, and then enter a name and choose a location to save your workflow.

Now, when you open the workflow, it will automatically open Fetch and upload the items you specified to the server you specified.

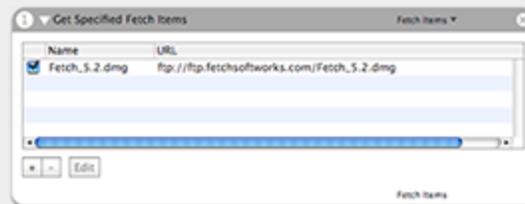
If the connection to the server requires a password, you will be prompted for it when you use the workflow. If you do not want to be prompted for the password, you should save the password for the connection in your [keychain](#).

If you want to edit your workflow later, drag the workflow's icon to the Automator application icon.

## Downloading files workflow

Follow these steps to create an Automator workflow that downloads specific files or folders from a server to your Macintosh. This can be useful if you frequently download the same set of files, or if you want to set up [scheduled downloads using iCal](#).

1. Open Automator (located in your Applications folder). A new workflow window should open automatically. If it does not, choose **File > New**.
2. In the new workflow window, click the Fetch icon in the Applications list.
3. Drag a Get Specified Fetch Items action from the Action list to the workflow area on the right side of the window.



4. In the Get Specified Fetch Items action, click the Add (+) button to add a remote file or folder to the list of items to download. When you click the Add button, a dialog similar to Fetch's New Connection dialog will appear.
5. Enter the connection information for the server where the file is stored, and a path to the item. (A path is the description of a file or folder's location that includes the folders the item is in, for example, "www/pictures/dog.jpg" is the path to the file "dog.jpg" that is in the folder "pictures" that in turn is in the folder "www".)
6. Use the Add (+) button to add as many files or folders as you want.
7. Next, drag a Download Fetch Items action from the Action list to the workflow area, placing it after the first action.



8. In the Download Fetch Items action, click the pop-up menu to choose the folder on your Macintosh where you want the files to be saved. If you do not want to use any of the listed folders, choose **Other** to select any folder on your Macintosh.
9. Finally, save the workflow as an application. Choose **File > Save As**. In the Save dialog, choose **Application** from the **File Format** pop-up menu, and then enter a name and choose a location to save your workflow.

Now, when you open the workflow, it will automatically open Fetch and download the items you specified to your Macintosh.

If the connection to the server requires a password, you will be prompted for it when you use the workflow. If you do not want to be prompted for the password, you should save the password for the connection in your [keychain](#).

If you want to edit your workflow later, drag the workflow's icon to the Automator application icon.

### **More information and more examples**

The two examples above are very simple examples of how you can use Automator with Fetch, using only a few of Fetch's Automator actions.

Other Fetch Automator actions let you rename, delete, or move files around on a server; set the permissions of items on a server; and filter items so that you download only items that meet a certain criterion and ignore other items. (There is also a Finder action for filtering Finder items, so that you upload only items that meet a certain criterion.)

For more information about each Fetch Automator action, see the [Automator actions](#) help topic.

For more examples of things you can do with Automator and Fetch using slightly more complex workflows, see the [more examples of using Automator With Fetch](#) help topic. It includes instructions of how to create workflows that batch change permissions of files on a server, download all files on a server that were created today, and more.



## [Fetch Help](#) > [Using Fetch](#) > More examples of using Automator with Fetch

Fetch includes a number of actions for Mac OS X's Automator, allowing you to create simple workflows that can automate a variety of Fetch tasks.

This topic describes some more examples of time-saving things you can do with Automator and Fetch. If you are not familiar with Automator, you may want to start by reading the [introduction to using Automator with Fetch](#) help topic, which also includes simple example workflows for uploading or downloading specific files.

For more details about each action, see the [Automator actions](#) help topic.

[Advanced Upload Droplet](#)  
[Control-click In the Finder to Upload](#)  
[Batch Change Permissions](#)  
[Batch Rename Files](#)  
[Multiple Server Upload Droplet](#)  
[Download All Files Created Today](#)  
[Delete All Old and Large Files](#)  
[Print a PDF to a File Server](#)  
[Setting Up a Watch Folder](#)

### Advanced Upload Droplet

Create a workflow that consists of an Upload Files action, configured for your favorite upload location, plus any actions you want to take after uploading the files — if you want to automatically change the files' permissions or rename the files after uploading them, add a Set Fetch Item Permissions or a Rename Fetch Items action after the Upload Files action.

Then save the workflow as an application: in Automator, choose **File > Save As** and choose **Application** from the **File Format** pop-up menu.

When you drag files and folders to the droplet, they'll be uploaded to your server automatically and modified according to the other actions that you included.

If you simply want a droplet to upload files to a specific server, and do nothing more, creating a droplet shortcut may be an easier solution than using Automator. See the [droplet shortcut](#) help topic for more information.

### Control-click In the Finder to Upload

If you're using Mac OS X 10.6 or later, create a new workflow, choosing **Service** from the template selector. Add an Upload Files action to the workflow and configure it for your favorite upload location. Then save the workflow.

If you're using Mac OS X 10.4 or 10.5, create a workflow that consists of an Upload Files action, configured for your favorite upload location. Then save the workflow as a Finder plug-in: in Automator, choose **File > Save As Plug-In**, and choose **Finder** from the **Plug-in for** pop-up menu.

Now when you Control-click on files and folders in the Finder, you can choose your your saved workflow from the Automator submenu of the contextual menu to upload your files automatically (on Mac OS X 10.5, after you Control-click, choose

the More submenu, then the Automator submenu). If you want to automatically change the files' permissions or rename the files after uploading them, add a Set Fetch Item Permissions or a Rename Fetch Items action after the Upload Files action.

## Batch Change Permissions

Create a workflow that consists of a Get Selected Fetch Items action, followed by a Set Fetch Item Permissions action. Then select the files whose permissions you want to set in a Fetch transfer window, and run the workflow. While you can batch change permissions using the [info window](#), you cannot set file and folder permissions independently when you do that; with the Set Fetch Item Permissions action, you can.

If you want to recursively change permissions (that is, change the permissions for all the files and folders inside the items you've selected) insert a Get Fetch Folder Contents action before the Set Fetch Items Permissions action, and check the "Repeat for each subfolder found" checkbox. Use with caution, since setting permissions is not undoable.

## Batch Rename Files

Create a workflow that consists of a Get Selected Fetch Items action, followed by a Rename Fetch Items action. Select the renaming options you want in the Rename Fetch Items action. Then in a Fetch transfer window, select the files whose names you want to set, and run the workflow. Use with caution, since renaming is not undoable.

## Upload Files to Multiple Servers Droplet

If you want to upload the same set of files to several different servers, you can create an Automator workflow that uses variables to simplify the task. The workflow stores the list of files you want to upload in a variable, so that you can recall the list of files repeatedly (because Automator will not store the list after the first action otherwise).

This workflow only works on Mac OS X 10.5 or later, because Automator variables are only available on Mac OS X 10.5 or later. Due to a problem with Automator on Mac OS X 10.5, you must include an extra action at the beginning of the workflow make it work.

Create a workflow with the following actions:

1. Sort Finder Items – You only need to add this action on Mac OS X 10.5 to force the Set Value of Variable action to work correctly; the files don't actually need to be sorted. It doesn't matter how you sort the items.
2. Set Value of Variable – This will store a reference to the files that were dropped on the workflow (you can give the variable any name you like, e.g. "Dropped Items").
3. Upload Files – Configure for the first server you want to upload the files to.
4. Get Value of Variable – From the pop-up menu, choose the variable

you specified in step 2. This will recall the reference to the dropped files.

5. Upload Files – Configure for the next server you want to upload the files to.

Repeat steps 4 and 5, adding as many "Get Value of Variable" and "Upload Files" actions as necessary, for all the servers you want to upload to.

Then save the workflow as an application: in Automator, choose **File > Save As** and choose **Application** from the **File Format** pop-up menu.

When you drag files and folders to the droplet, they'll be uploaded to all the servers automatically.

## Download All Files Created Today

If you have a folder on your server where clients or co-workers put new files for you, you can create a workflow that will check that folder for new files and download them all. Create a workflow with the following actions:

1. Get Specified Fetch Items
2. Get Fetch Folder Contents
3. Filter Fetch Items: Whose "Item Type is File"
4. Filter Fetch Items: Whose "Modification Date Is Today"
5. Download Fetch Items

Configure the Get Specified Fetch Items action with the server and folder that you want to check, and in the Download Fetch Items, choose the local folder where you want the files downloaded to. If you'd rather download files that were modified in the last week, change the second Filter Fetch Items action's options to be "Modification Date Is within the last week," or any other time period you prefer.

The Get Fetch Folder Contents action gets a list of the contents of the folder. The "Item Type is File" filter action ensures that only files will be downloaded, not the entire folder being checked, or any folders in it.

## Delete All Old and Large Files

If you have a folder on your server that often fills up with junk, you could create a workflow that will look in that folder and delete any old, large files to free up space on the server. Create a workflow with the following actions:

1. Get Specified Fetch Items
2. Get Fetch Folder Contents
3. Filter Fetch Items: Whose "Item Type is File"
4. Filter Fetch Items: Whose "Modification Date Is More Than a Month Ago"
5. Filter Fetch Items: Whose "Size Is greater than 500 KB"
6. Delete Fetch Items

Configure the Get Specified Fetch Items action with the server and folder that you want to remove files from. Adjust the time and size values of the second and third filter items as you prefer.

The Get Fetch Folder Contents action gets a list of the contents of the folder. The first Filter Fetch Items action that filters for files is necessary to prevent you from deleting an entire folder accidentally (because it's unlikely you want this workflow to do that).

Since deleting items is not undoable, be very careful when using a workflow similar to this one. If you check the "Ask before deleting" checkbox in the Delete Fetch Items action, you will have a chance to see which files will be deleted and cancel the action.

## Print a PDF to a File Server

If you're using Mac OS X 10.6 or later, create a new workflow, choosing **Print Plugin** from the template selector. Add an Upload Files action to the workflow, and configure it for the server and remote folder where you want the PDFs to be uploaded. Then save the workflow.

If you're using Mac OS X 10.4 or 10.5, create a workflow that consists of an Upload Files action, configured for your the server and remote folder where you want the PDFs to be uploaded. Then save the workflow as a Print plug-in: in Automator, choose **File > Save As Plug-In**, and choose **Print Workflow** from the **Plug-in for pop-up** menu.

Now you can create and upload a PDF document to that server from the Print dialog of any application. In the standard Mac OS X Print dialog, click the PDF button, and choose your saved workflow from the PDF menu. The PDF is created and uploaded automatically. (The PDF uploaded to the server will have the same name as the file you're printing, followed by the .pdf extension.)

## Setting Up a Watch Folder

If you'd like files to be automatically uploaded when you move or save them to a specific folder on your Mac — often referred to as a watch folder — you can use Fetch's Automator support together with a feature of Mac OS X called folder actions to set one up. See the [setting up a watch folder](#) help topic for more information and step-by-step instructions.

## [Fetch Help](#) > [Using Fetch](#) > Time-saving ways to use Fetch

Use the [Edit](#) button or command to make quick changes to web pages or server configuration files without having to separately download and re-upload the files.

Configure Fetch to [Edit files instead of downloading them](#) when you double-click on them.

Use Fetch to [rename](#), move, delete, and [change the permissions](#) of files on a web server.

Update your website with just a double-click by creating a [mirror document](#). See the [Mirroring a local folder to a remote folder](#) help topic for step-by-step instructions.

Set up [WebView](#) so that you can view files in a web browser by clicking the **WebView** button, and copy the web address of those files.

Display files (images, movies, sounds, HTML, text) without leaving Fetch by using the [Quick Look](#) button or command (or [View](#) on Mac OS X 10.4).

[Create shortcuts](#) to commonly used files and folders to make downloading them a simple process.

[Create a default shortcut](#) to have the New Connection dialog filled out automatically. You can also make an existing shortcut the default by using the **Default shortcut** pop-up menu in the [General Preferences](#) pane.

Save [droplet shortcuts](#) so that you can upload files by dragging files to your favorite servers' icons in the Finder.

Control-click (or right-click with a 2-button mouse) on file list items and shortcuts to quickly access commonly used commands.

Use the [Get Info](#) command and [info window](#) to find out up how much space your files are taking on your server, or how close you are to your server storage limit. When you select a set of files and folders, choose **Get Info**, and click the **Calculate** button, Fetch calculates the disk space taken up not only by those items, but by the items inside the folders too.

Select multiple items and click the [Get Info](#) button to [set the permissions](#) for all of them at once.

Use the [StuffIt X Archive](#), [Zip Archive](#), and [Gzipped Tar Archive](#) upload formats to upload files as compressed archives automatically.

Use iCal along with Fetch's Mirror command or Automator support to [schedule automatic transfers](#).

Record an AppleScript to automate routine tasks, such as deleting old web logs. To record an AppleScript, start Script Editor, click the **Record** button, switch to Fetch, and then perform the task. When you're done, return to Script Editor, click **Stop**, and save your AppleScript.

Also see the [introduction to using Automator with Fetch](#) and [more examples of using Automator with Fetch](#) help topics for ways to save time with Fetch's [Automator actions](#).

This topic describes some ways to do things in Fetch that may not be immediately obvious.

Command-double-click a remote folder in a [file list](#) to open a new transfer window displaying the contents of that folder.

To move files and folders up one level (to the parent folder) without opening a new window or connection, add the [Parent](#) button to the [transfer window toolbar](#), and then drag items to the Parent button icon.

To [Get](#), [Edit](#), or [Delete Items](#) by typing in the name or path of an item, hold down the Shift key when choosing the command from the [Remote](#) menu or when clicking on the equivalent toolbar button.

Control-click on a transfer window to display the file list contextual menu. Choose the [Clone Window](#) command to open a new connection to the same server. Drag files and folders between the two transfer windows to move them between folders on the server.

You can [upload files using the Copy and Paste commands](#), or use [Copy and Paste to copy files from one folder or server to another](#).

You can open a connection to a shortcut or a recent connection by clicking and holding down on the Fetch icon in the Dock until the Dock menu appears, and then choosing an item from either the **Shortcuts** or **Recent Connections** submenu.

On laptops whose trackpads support gestures, in Fetch you can swipe left to go back to previously viewed folder, swipe up to go to the parent folder, and swipe down to open the currently selected folder.

When you choose a [Bonjour](#) server from the **Shortcuts** menu, Fetch automatically tries to log you in to the server with your current Mac OS X username. To use a different username, hold down the Option key when choosing the server from the **Shortcuts** menu. This will display the [New Connection dialog](#) filled out with the information for connecting to the Bonjour server. Edit the username, and then click **Connect**.

Option-double-click a shortcut in Fetch Shortcuts or another shortcut list window to bring up the [New Connection dialog](#) filled out with information from the shortcut.

Option-Shift-double-click a shortcut in Fetch Shortcuts or another shortcut list window to open the [Edit Shortcut dialog](#).

Command-double-click a shortcut in Fetch Shortcuts or another shortcut list window to open it in a new window, even if a transfer window to that server is already open.

Press Command-] or Command-[ to cycle through the shortcuts in the [New Connection dialog](#). You can cycle through the recent connections while in the dialog by pressing Command-Option-] or Command-Option-[. (In the French localization, these keyboard shortcuts are Command-), Command-(, Command-Option-), and Command-Option-(, respectively.)

Command-click the proxy icon or title in a [transfer window](#) title bar to show the path to the current folder; choose a folder from the path pop-up to make it the current folder. Drag the proxy icon to a shortcut list window to create a shortcut to the current folder, to the Finder to create a droplet shortcut, or to a text editor to copy the folder's Fetch address (URL) or web address if you've set up [WebView](#) for that server.

Command-click the proxy icon or title in an [info window](#) title bar to show the path to the items described in the window; choosing a folder from the path pop-up shows that folder in a transfer window. Drag the proxy icon to a shortcut list window to create a shortcut to the items, to the Finder to create a droplet shortcut (for folders) or a shortcut document (for files), or to a text editor to copy the items' Fetch address (URL) or web address if you've set up [WebView](#) for that server.

Also see the [time-saving ways to use Fetch](#), [keyboard shortcuts](#) and [drag-and-drop](#) help topics for additional tips.



## [Fetch Help](#) > [Using Fetch](#) > Sending a transcript to Fetch Softworks

Sometimes, when providing support for Fetch, we may ask you to send a transcript to Fetch Softworks to help us troubleshoot the problem. Here's how to do that:

1. Attempt to reproduce the problem.
2. Choose **Window > Fetch Transcript**.
3. Choose **Edit > Select All**.
4. Choose **Edit > Copy**.
5. Open a new email message addressed to [support@fetchsoftworks.com](mailto:support@fetchsoftworks.com) in your email program.
6. Give the message a descriptive subject line, and write a short description of the problem in the message body.
7. Click the body of the message, and choose **Edit > Paste**.
8. The transcript, which will be rather long, should appear in your message.
9. Send the email.

See the [Fetch Transcript](#) help topic for more information about the transcript.

## [Fetch Help](#) > [Using Fetch](#) > **Allowing Fetch to work with the Mac OS X firewall (Mac OS X 10.5 or later)**

If you are having trouble connecting to a server, one of the possible reasons is that it requires active FTP connections, and you have the Mac OS X firewall enabled. Active FTP connections cannot work successfully with any firewall. For more information about active and passive mode connections, see the [active and passive FTP](#) help topic.

On Mac OS X 10.5 Leopard or later, it is possible to tell the Mac OS X firewall to allow an exception for Fetch while continuing to shield your computer from other network communications. Unfortunately, this feature is not available on Mac OS X 10.4; on that version of Mac OS X, you will have to turn the firewall off temporarily.

Note that you do not usually need to change your firewall settings in order for Fetch to work successfully. If you are experiencing problems connecting with a server, you should first check for other, more common causes, such as:

- Verify that you have the correct hostname, username, and password.
- Verify that the server is running.
- Make sure you are trying passive FTP connections first, which are more compatible with firewalls than active FTP connections. Choose to **Fetch > Preferences**, click the [General](#) tab, and make sure the **Use passive mode transfers (PASV)** checkbox is checked.

If none of these help, then you should see if your Mac OS X firewall could be causing the problem. If your Mac OS X firewall is not turned on, it is not the cause of the problem, and changing its settings will not help.

Configuring the Mac OS X firewall is slightly different on Mac OS X 10.5 and 10.6. Follow the appropriate directions below, depending on which version of Mac OS X you are using.

### **Mac OS X 10.6**

1. Go to the Apple menu, and choose **System Preferences**.
2. Click the **Security** icon (or choose **View > Security**).
3. Click the **Firewall** tab.
4. If it states **Firewall: Off**, stop here. Your Mac OS X firewall is not enabled. You should contact the server administrator (the person or people who maintain the server you are trying to connect to) for further advice.

If your firewall is enabled, continue:

5. Click the **Advanced...** button. (You may need to first click the lock icon in the lower left of the window and enter your Mac OS X password to make this change.)
6. If the **Block all incoming connections** box is checked, uncheck it.
7. Click the Add (+) button (beneath the large white rectangle).
8. A dialog to select an application appears. Find the Fetch application (usually located in your Applications folder), and click **Add**.
9. The Fetch icon should now be listed in the Firewall preferences window, saying "Allow incoming connections."

10. Click **OK** and close the Firewall System Preferences window.

## Mac OS X 10.5

1. Go to the Apple menu, and choose **System Preferences**.
2. Click the **Security** icon (or choose **View > Security**).
3. Click the **Firewall** tab.
4. If the **Allow all incoming connections** button is selected, stop here. Your Mac OS X firewall is not enabled. You should contact the server administrator (the person or people who maintain the server you are trying to connect to) for further advice.

If your firewall is enabled, continue:

5. Click the **Set access for specific services and applications** button.
6. Click the Add (+) button (beneath the large white rectangle).
7. A dialog to select an application appears. Find the Fetch application (usually located in your Applications folder), and click **Add**.
8. The Fetch icon should now be listed in the Firewall preferences window, saying "Allow incoming connections."
9. Close the Firewall System Preferences window.

From now on, the Mac OS X firewall will allow active FTP connections made by Fetch to succeed.

Note that if there is a firewall elsewhere on your network – for instance, one installed by your company to protect all the computers in your office, or one on your router – changing the settings of the Mac OS X firewall will not help. You will need to talk to your company's network administrator, or consult your router's manual instead.

Related topics:

- [Active and passive FTP](#)



## [Fetch Help](#) > [Using Fetch](#) > **Converting line endings when uploading with SFTP**

Unlike FTP servers, most SFTP servers do not support automatic translation of line endings in text files when uploading, so normally Fetch leaves line endings unchanged when you upload text files using SFTP (that is, the line endings remain the same as they were in the file on the Macintosh, regardless of whether that is appropriate for the server the files are uploaded to).

If you want to force line endings to be converted to a certain style when uploading text files to an SFTP server, you can use a "secret preference" to tell Fetch which kind of line ending to use. To do this:

1. Quit Fetch.
2. Open the Terminal application (in the Utilities folder of the Applications folder).
3. Type (or copy from here, and paste) the following line:  
`defaults write com.fetchsoftworks.Fetch SFTPUploadTextEOLStyle -int x`
4. Replace the "x" at the end with one of the following numbers, depending on what you'd like:
  - 0: Leave line endings unchanged (the default)
  - 1: Change line endings to CR (a single carriage return, common on Mac OS 9 and older)
  - 2: Change line endings to LF (a single linefeed, common on Mac OS X and UNIX)
  - 3: Change line endings to CRLF (a carriage return followed by a linefeed, common on Windows and DOS)
5. Open Fetch again.

This setting does not affect the conversion of line endings when uploading with FTP; normally that is handled automatically by the server.

This setting does not affect the conversion of line endings when downloading with SFTP or FTP; you can change that by using the **Text file line endings** pop-up menu in the [Download Preferences](#) pane.



## [Fetch Help](#) > [Using Fetch](#) > **Displaying VMS version numbers in file lists**

Normally, when you connect to a VMS file server, Fetch does not display version numbers after the file names. However, on some VMS servers, you need the version number displayed in order to delete files or to indicate a specific version of a file to transfer.

You can use a "secret preference" to tell Fetch to display VMS version numbers in file lists. To do this:

1. Quit Fetch.
2. Open the Terminal application (in the Utilities folder of the Applications folder).
3. Type (or copy from here, and paste) the following line:  
`defaults write com.fetchsoftworks.Fetch ShowVMSVersionNumbers -bool true`
4. Open Fetch again.
5. You may need to [refresh](#) your file lists to see the version numbers.

To stop displaying VMS version numbers in file lists, repeat the above steps, except use `-bool false` at the end of the line instead.



[Fetch Help](#) > [Using Fetch](#) > **Importing Interarchy bookmarks**

You can import your favorites from Interarchy into Fetch using the link below. Your Interarchy bookmarks will appear in the [Fetch Shortcuts](#) window.

[Import my Interarchy bookmarks for me](#)

If you do not have any Fetch shortcuts the first time you open Fetch 5.6, Fetch will automatically try to import your Interarchy bookmarks.

Interarchy bookmarks that use a connection type not supported by Fetch (such as HTTP bookmarks) will not be imported.



[Fetch Help](#) > [Using Fetch](#) > **Importing Transmit favorites**

You can import your favorites from Transmit into Fetch using the link below. Your Transmit favorites will appear in the [Fetch Shortcuts](#) window.

[Import my Transmit favorites for me](#)

If you do not have any Fetch shortcuts the first time you open Fetch 5.6, Fetch will automatically try to import your Transmit favorites.

Transmit favorites that use a connection type not supported by Fetch (such as WebDAV) will not be imported.



## [Fetch Help](#) > [Using Fetch](#) > **Moving keychains and passwords to another Macintosh**

When you are upgrading to a new Macintosh, or if you use multiple Macintosh computers, you may want the same Fetch shortcuts and their passwords on each computer.

[Moving your shortcuts](#) is fairly straightforward. However, to provide the best security, Fetch stores its passwords separate from the shortcuts file; passwords are stored in the Mac OS X [keychain](#). Unfortunately, this means when you move your shortcuts file from one Macintosh to another, all the information about your shortcuts except the passwords is moved; and moving keychains from one Macintosh to the other is not as straightforward as moving the shortcuts.

The simplest solution, especially if you just have a few passwords, is to re-enter your passwords on the new Macintosh (where they will be saved in that Macintosh's keychain). See the Fetch [Password FAQ](#) help topic for information about how to reveal a shortcut's password using the Keychain Access application.

Otherwise, there are a few options for moving keychain information between two Macintosh computers:

- If you are a MobileMe (.Mac) subscriber, you can use [MobileMe \(.Mac\) syncing](#) to sync the two keychains, which will copy passwords that are not in one keychain to the other. This is the best option if it is available to you.
- The [Mac OS X Migration Assistant](#) will move keychain files between computers, although usually the Migration Assistant is most useful if you have not moved your files to the other Macintosh yet.
- Or you can follow the instructions below to move your keychain from one Macintosh to another, and then move the passwords from one keychain to another. Note that you will need to know your login (or administrator) password from the old Macintosh in order to move the rest of your passwords.

To move your keychain from one Macintosh to another and then move the passwords from one keychain to another (these instructions only apply to Mac OS X 10.4):

1. Locate your keychain file on your old Macintosh. One way to do this is to choose **Go > Home** in the Finder of the old Macintosh, double-click on the Library folder, and then double-click on the Keychains folder. You should see a file named login.keychain inside that folder.
2. Duplicate or make a copy of the keychain file.
3. Rename the duplicate (or copy) of the old keychain file to something like "old Mac.keychain" — be sure the name of the file still ends in ".keychain".
4. Copy the old keychain to the new Macintosh.
5. Put the old keychain in the Keychains folder on the new Macintosh. To find the Keychains folder, choose **Go > Home** in the Finder, double-click on the Library folder, and then double-click on the Keychains folder.
6. Open the Keychain Access application, which is in the Utilities folder inside your Applications folder.
7. Choose **Edit > Keychain List**.
8. Click the Plus (+) button.
9. Locate the old keychain file, select it, and click the Open button.
10. Click OK to close the Keychain List window.
11. In the list of keychains on the left, select the keychain from the old Macintosh.
12. Find a password you want to move to your new Macintosh computer's keychain. (The

passwords are usually listed by the name of the server they are for; Fetch passwords will have a kind of "Fetch password" or "Internet password.")

13. Drag it to the "login" keychain entry in the keychain list on the left. The first time you move a Fetch password, you will be prompted to enter a password twice: one to unlock the old keychain, and one to give permission to modify the keychain item — both times, you should enter the login or administrator password from the old Macintosh. For subsequent Fetch passwords, you will only be asked to enter your old login password once.
14. Repeat steps 12–13 until all your passwords have been relocated in the new keychain.

Now when you use the shortcuts you copied to the new Macintosh, your password should automatically be retrieved from the keychain.

If you are upgrading from a previous version of Fetch on the same computer, you do not need to import your old shortcuts and passwords. Fetch will do that automatically.

Related topics:

- [Moving your shortcuts to another Macintosh](#)
- [Keychain support](#)
- [Shortcuts](#)



Fetch supports authentication via SSH public/private keys or ssh-agent when connecting to SFTP servers.

To use SSH public/private keys:

1. Choose **File > New Connection**.
2. Enter the server's hostname and your username.
3. Choose **SFTP** from the **Connect using** pop-up menu.
4. In the password field, enter a dummy password — just type anything (but don't leave it blank).
5. Click the **Connect** button.
6. If your SSH key is protected by a passphrase, Fetch will prompt you for it.
7. Fetch will use your SSH key for authentication and make the connection.

To authenticate via ssh-agent:

1. Enter your password in ssh-agent.
2. Choose **File > New Connection**.
3. Enter the server's hostname and your username.
4. Choose **SFTP** from the **Connect using** pop-up menu.
5. In the password field, enter a dummy password — just type anything (but don't leave it blank).
6. Click the **Connect** button. Fetch will use ssh-agent for authentication and make the connection.

Note that Fetch does not provide support for setting up ssh-agent or SSH public/private keys; you must do that outside of Fetch.

If you'd like to avoid entering a dummy password every time, check the **Add to keychain** checkbox beneath the password field after you enter a dummy password. For all future connections, Fetch will automatically retrieve the dummy password from the [keychain](#), and then it will go on to use your SSH keys.

 [Fetch Help](#) > **Frequently Asked Questions (FAQ)**

The following topics provide answers to some of the most frequently asked questions about Fetch. If there's a question that you think should be addressed here, please send it to [faq@fetchsoftworks.com](mailto:faq@fetchsoftworks.com). Thank you!

[Using Fetch FAQ](#)

[Troubleshooting FAQ](#)

[Fetch Features FAQ](#)

[Password FAQ](#)

[Licensing & Registering FAQ](#)

[Other FAQ](#)



Here are answers to commonly asked questions about using Fetch.

1. [What do I enter in the Hostname, Username and Password fields?](#)
2. [How do I make Fetch start up with the information for my server or website?](#)
3. [How do I create, delete or change shortcuts?](#)
4. [How do I upload more than one file at a time?](#)
5. [How do I make downloaded files open in the correct application?](#)
6. [How do I connect to an FTP or SFTP server on a non-standard port?](#)
7. [How do I disconnect from a server?](#)
8. [How can I see the modification times for files?](#)
9. [How do I use my shortcuts from a previous version of Fetch in Fetch 5.6?](#)
10. [How do I move my shortcuts from one Macintosh to another?](#)
11. [How do I upload CGI scripts in ASCII format?](#)

1. *What do I enter in the Hostname, Username and Password fields?*

You need to enter the hostname (something like "home.example.com" or "ftp.example.net") of the server you want to transfer files from or to, and the username (something like "smith97" or "billsbakery") and password for your account on that server. If this is an account you've purchased from a Internet service provider, that provider should be able to tell you what to enter. See the [What do I enter in the hostname, username, and password fields?](#) help topic for more information.

2. *How do I make Fetch start up with the information for my server or website?*

You can make Fetch automatically fill out the Hostname and Username fields in the New Connection dialog by using a default shortcut. Fill out your server's information in the [New Connection dialog](#), choose **Make Shortcut** from the Shortcuts  pop-up menu to the right of the Hostname field, and check the **Make this the default shortcut** box. See the [making Fetch start with information for your server or website](#) help topic for more detailed steps.

3. *How do I create, delete or change shortcuts?*

To create a shortcut, use the **Make Shortcut** command in the Shortcuts  pop-up menu in the New Connection dialog, or the **Shortcuts > New Shortcut** command, or drag an item from a transfer window to the Fetch Shortcuts window. For more detailed steps, see the [creating a shortcut](#) help topic.

To delete a shortcut, choose **Shortcuts > Show Shortcuts**, select the shortcut in the Fetch Shortcuts window, and choose **Edit > Delete** or press the Delete key. To change a shortcut, choose **Shortcuts > Show Shortcuts**, select the shortcut in the Fetch Shortcuts window and choose **Shortcuts > Edit Shortcut**. For more information, see the [deleting or changing a shortcut](#) help topic.

4. *How do I upload more than one file at a time?*

There are several different ways to upload multiple files (or entire folders) with one

command:

- Either click the **Put** button in the transfer window toolbar or choose **Remote > Put**, and then either select a folder, or hold down the Command or Shift keys while clicking to select multiple files
- Drag file and folder icons from the Finder to a Fetch transfer window
- Create a [droplet shortcut](#) for your server and drag files and folders to it in the Finder or Dock

If you want to only upload new or changed files in a folder, you may want to use the [Mirror](#) command in the **Remote** menu. See the [Mirror window](#) or [mirroring a local folder to a remote folder](#) help topics for more information.

#### 5. How do I make downloaded files open in the correct application?

Normally Fetch uses the extension at the end of a file's name to set the application that a file will open with. Usually Fetch will choose the same application that the Finder uses to open files with a specific extension. However, you can change which application will open which extension in the [Transfer Options](#) pane of the [info window](#) in Fetch — see the [changing the application a file opens with](#) help topic.

Also, if files have been encoded in a special format that preserves Macintosh information, such as BinHex, MacBinary, AppleSingle, Zip Archive, or Stuffit X Archive, the files will open in the application originally specified for those files when they were uploaded. If the setting was different on the computer that uploaded those files than on yours, they may not open in the application you expect.

#### 6. How do I connect to an FTP or SFTP server on a non-standard port?

Click the disclosure  button in the [New Connection dialog](#) to reveal the field for entering a non-standard port.

#### 7. How do I disconnect from a server?

To disconnect from a server, simply close the Fetch transfer window for that server. If you do not want to close the transfer window, leave it open and Fetch will automatically disconnect from the server after approximately 5 minutes (the status pane will display "Disconnected" once this happens). Fetch will automatically reconnect if you decide to work with the transfer window again.

#### 8. How can I see the modification times for files?

While it does not show the modification times by default, you can resize the Date column in the [file list](#) to show the time as well as date. To resize a column in the file list, click the vertical line between the titles of two columns and drag.

You will probably need to make another column narrower in order to provide space to expand the Date column; so first, make another column (such as the Name column) narrower, and then make the Date column wider. If you feel you do not have enough space to make all columns as wide as you like, make the entire window wider and then resize the columns.

You can also use the [Get Info](#) command to see the modification time of a single file.

#### 9. How do I use my shortcuts from a previous version of Fetch in Fetch 5.6?

The first time you open Fetch 5.6, it will automatically read and upgrade your shortcuts from your previous version of Fetch. However, Fetch 5.6 stores shortcuts in a different format than some previous versions of Fetch, so while you can continue to use Fetch 5.6 and an earlier version of Fetch on the same Macintosh, you may not see changes you make to the Fetch 5.6 shortcuts in the previous versions of Fetch, and vice versa.

Fetch 5.1 and later store your shortcuts in a file named "com.fetchsoftworks.Fetch.Shortcuts.plist" in the Preferences folder in the Library folder in your home folder. Fetch 5 and Fetch 4.0.3 both stored shortcuts in a file named "Fetch Shortcuts" in that same location. (After you upgrade from one of those older versions, you will have both files — the older file is saved both as a backup, and in case you want to continue to use the earlier version of Fetch.)

If you are setting up a new Macintosh or new hard drive, or upgrading to Mac OS X, you can copy the com.fetchsoftworks.Fetch.Shortcuts.plist or Fetch Shortcuts file from the old disk to the right place on your new Macintosh. (On OS 9, the Fetch Shortcuts file is located in the Preferences folder in your System Folder.) See the [moving shortcuts to another Macintosh](#) help topic for more information.

*10. How do I move my shortcuts from one Macintosh to another?*

If you are upgrading to a new Macintosh, or use multiple Macintosh computers and would like to have the same Fetch shortcuts on each computer, see the [moving shortcuts to another Macintosh](#) help topic for detailed steps on moving your shortcuts to another computer.

*11. How do I upload CGI scripts in ASCII format?*

Fetch refers to ASCII format as **Text** format. To upload files in **Text** format, choose **Text** from the [Format](#) submenu of the [Remote menu](#), and then choose the **Remote > Put** menu command or click the **Put** button in the transfer window toolbar.



These are some common problems when using Fetch and what to do about them.

1. [The server says "Incorrect Password" but I know the password is correct.](#)
2. [Fetch connects, but the file list never appears and the Fetch dog runs and runs.](#)
3. [The Fetch dog runs and runs but never connects.](#)
4. [All of my file appears to transfer \(the progress bar fills the entire circle\), but at the end Fetch stalls and the transfer never finishes.](#)
5. [I uploaded an updated webpage but my browser still shows the old one.](#)
6. [The file list does not include a file that I know was just put there.](#)
7. [The server says "Invalid PORT Command" or "Can't build data connection. Connection refused."](#)
8. [How do I stop Fetch from expanding every Zip and Stuffit archive?](#)
9. [Why isn't TextEdit listed as an option for editing HTML files \(webpages\)?](#)
10. [Get Info shows the permissions of a file or folder I just uploaded as blank or 000.](#)
11. [Downloaded files have the wrong icon.](#)
12. [Why can't I make files open or edit in the correct application?](#)
13. [The filenames in the file list don't look right — there is size and date information in the name column, and the \*\*Get\*\* button doesn't work.](#)
14. [I can connect to a server on my local network, but not to Internet FTP or SFTP servers.](#)
15. [Why is uploading files slower than downloading files?](#)

1. *The server says "Incorrect Password" but I know the password is correct.*

Please see [I received an error saying "login incorrect" or "rejected password" in the Solving Problems with Connecting](#) help topic for suggestions.

2. *Fetch connects, but the file list never appears and the Fetch dog runs and runs.*

Toggle the setting of the **Use passive mode transfers (PASV)** checkbox in the [General Preferences](#) pane, and try connecting again. If toggling the **Use passive mode transfers (PASV)** checkbox didn't help, try disabling the OS X firewall in the **Sharing** pane of **System Preferences**, if it is enabled. On Mac OS X 10.5 and later, you can [configure the firewall to allow an exception for Fetch](#). For more information about Fetch, firewalls, and active vs. passive FTP, see the [active and passive FTP](#) topic.

3. *The Fetch dog runs and runs but never connects.*

Usually this means that the server you are trying to connect to is not currently operating. Another possibility is that there is a network outage or a firewall somewhere between your computer and the server. If the problem persists, confirm that you are connecting to the right hostname, and check with your network administrator about firewalls. You can use the ping and traceroute functions of the Network Utility application (which you can find by going to your Applications folder, and then the Utilities folder) to check the network route between your computer and the server.

4. *All of my file appears to transfer (the progress bar fills the entire circle), but at the end Fetch stalls and the transfer never finishes.*

The most common cause of this is that you are behind a NAT device (such as a router or wireless base station) and the transfer has really finished, but the NAT device has blocked the information telling Fetch that it's done. Your transfer is done, just click the **Stop**  button to clear the progress indicator.

To resolve these problems, our recommendations are to:

1. Make sure the **Use passive mode transfers (PASV)** checkbox in the [General Preferences](#) pane is checked, and try again;
2. Download and try [the latest version of Fetch](#) and see if it fixes the problem for you;
3. If the problem still happens with the latest version of Fetch, try a different FTP client (such as Interarchy, Transmit, or Vicomsoft FTP Client) to check whether the problem is specific to Fetch or more general—if the problem only happens with Fetch, please contact us and let us know so we can investigate;
4. If the problem happens with all FTP clients, contact the server administrator or network administrator, and see if they can reproduce the problem under similar conditions.

5. *I uploaded an updated webpage but my browser still shows the old one.*

Please see the [solving problems with websites](#) help topic for suggestions and possible solutions.

6. *The file list does not include a file that I know was just put there.*

Choose the **Refresh** command in the **View** menu; that will usually fix this problem. Fetch reuses file lists when they have not changed in a long while, which makes loading them quicker but also makes it possible to miss new changes.

7. *The server says "Invalid PORT Command" or "Can't build data connection. Connection refused."*

These problems can usually be resolved by checking the **Use passive mode transfers (PASV)** checkbox in the [General Preferences](#) pane. See the [active and passive FTP](#) topic for more information about this checkbox.

8. *How do I stop Fetch from expanding every Zip or Stuffit archive?*

Normally Fetch automatically expands Zip and Stuffit archives and other encoded or compressed files after downloading them. To stop Fetch from expanding or decompressing *all* files, uncheck the **Allow automatic decoding of downloaded files** checkbox in the [Download Preferences](#) pane.

You can also tell Fetch to stop automatically expanding just certain kinds of files. For example, if you want to stop Zip files from being expanded: Connect to your server, select a file whose name ends in ".zip", choose **Remote > Get Info**, and uncheck the **Automatically decode files like this** checkbox in the Transfer Options pane of the info window. To stop decoding of Stuffit archives, select a file whose name ends in ".sit" and follow the same steps. For more information, see the [decoding files automatically](#) help topic.

You can also follow these steps to in order make Fetch save the original Zip and

Stuffit archives (and other compressed files) after downloading and expanding them. For more information, see the [saving original archives files after expanding them](#) help topic.

9. *Why isn't TextEdit listed as an option for editing HTML files (webpages)?*

TextEdit, the text editor included with Mac OS X, is a very poor choice for editing webpages. While TextEdit appears to let you edit HTML files in a more natural manner because you don't have to edit the HTML code, it doesn't understand all HTML formatting. If you save your HTML file after editing it in TextEdit, it is nearly guaranteed that your webpage's links will be broken and you'll have lost some of the formatting.

Therefore, Fetch does not offer TextEdit directly as an editor choice for HTML files. If you are determined to use it for editing your HTML files, you can set it as the editor by choosing the **Other** menu item from the **Edit files like this with** pop-up menu in the [Transfer Options](#) pane of the [info window](#). But we strongly recommend against it.

Instead, we recommend that you consider using [TextWrangler](#), an excellent and free text editing program for the Macintosh. However, you do have to edit the HTML code directly with it. There are many other fine text editors for the Macintosh as well, such as BEdit, TextMate, SubEthaEdit, or CSSEdit to name a few.

There are also several good low-cost webpage editors such as iWeb, RapidWeaver, or Sandvox.

TextEdit is a perfectly good application for editing plain text files, but it is simply not a good HTML editor.

10. *Get Info shows the permissions of a file or folder I just uploaded as blank or 000.*

For just uploaded items on FTP connections, or newly created folders and files, Fetch may show all the permissions checkboxes as unchecked (a UNIX equivalent of "000") and the owner and group as "unknown". This is not the true state of the permissions. To see the true permissions, you must click the [Refresh](#)  button in the transfer window, and [Get Info](#) on the items again. On SFTP connections, you will see the correct permissions immediately, but you must refresh to see the true owner and group.

11. *Downloaded files have the wrong icon.*

If a certain kind of downloaded file does not have the icon you expect after downloading, try changing the application that kind of file is set to open with in the [Transfer Options](#) section of the [info window](#). See the [changing the application a file opens with](#) help topic for step-by-step instructions.

12. *Why can't I make files open or edit in the correct application?*

First, make sure you have followed the instructions for setting which application opens or edits the desired kind of file in the [changing the application a file opens with](#) or [changing the editor for a file](#) help topics.

However, if the application you want to use to open or edit a kind of file is not listed even though it is installed, or you choose it and Fetch does not remember your choice, or you choose one application and the files still open in a different application, you may be being affected by a problem in Mac OS X. Follow the

instructions in the [fixing problems choosing editors and helper applications](#) help topic to fix this problem.

*13. The filenames in the file list don't look right — there is size and date information in the name column, and the Get button doesn't work.*

If the Fetch file list is showing scrambled information, it may be that Fetch misunderstood the list of files the server sent to it. Choosing the **Refresh** command in the **View** menu will usually redraw the file list with the correct information. Otherwise, if you know the name of the file, you can hold down the Shift key and click the **Get** button and type in the file's name. If refreshing does not fix the problem, please [send a transcript to Fetch Softworks](#) so we can fix the problem in future versions of Fetch.

*14. I can connect to a server on my local network, but not to Internet FTP or SFTP servers.*

Check with your network or firewall administrator to see if you need to use a SOCKS gateway or proxy server to access Internet FTP or SFTP servers. If so, you can configure Fetch to work with the proxy server using the [Proxy Preferences](#) pane.

*15. Why is uploading files slower than downloading files?*

Usually this is normal, because most high-speed Internet connections, including cable modems and DSL, are asymmetric — they are designed to provide much better speed for downloading than uploading. For more information, see the [Why is uploading slower than downloading?](#) help topic.



Here are answers to commonly asked questions about using Fetch.

1. [What are the new features in Fetch 5.6?](#)
2. [Does Fetch support SFTP?](#)
3. [Does Fetch support FTP with TLS/SSL \(FTPS\)?](#)
4. [Does Fetch support MobileMe \(.Mac\) or WebDAV?](#)
5. [Does Fetch support challenge-response password systems?](#)
6. [Does Fetch support resuming uploads?](#)
7. [Is Fetch Section 508 compliant?](#)

1. *What are the new features in Fetch 5.6?*

If you're a regular user of Fetch upgrading from an earlier version, see either the [Fetch 5.6 for Fetch 5 Users](#) or the [Fetch 5.6 for Fetch 4 Users](#) help topic to learn about new features and changes in Fetch. You can also see the [summary of new features in Fetch 5.6](#) topic for a simple list of the major changes and improvements, or read the complete list of changes, new features, and fixes in the [Fetch 5.6 release notes](#).

2. *Does Fetch support SFTP?*

Yes. Choose **SFTP** from the **Connect using** pop-up menu in the [New Connection dialog](#). For more information, see the [SFTP](#) help topic.

3. *Does Fetch support FTP with TLS/SSL (FTPS)?*

Yes. Choose **FTP with TLS/SSL** from the **Connect using** pop-up menu in the [New Connection dialog](#). Fetch supports both AUTH TLS (also known as FTPES or Explicit SSL) and SSL connect (also known as Implicit SSL) connections. To use SSL connect (Implicit SSL), choose **FTP with TLS/SSL** and enter 990 in the Port field. For more information, see the [FTP with TLS/SSL](#) help topic.

4. *Does Fetch support MobileMe (.Mac) or WebDAV?*

No, sorry, Fetch does not support connecting or transferring files to MobileMe (.Mac) or other WebDAV servers as of Fetch version 5.6. You can use the Mac OS X Finder to connect to your iDisk and transfer files to your MobileMe (.Mac) account.

5. *Does Fetch support challenge-response password systems?*

Yes. If Fetch detects a One-Time Password challenge, it automatically computes the one-time password, based on the secret password entered by the user. If, for some reason, you need to compute the response yourself, enter "\*\*\*CHALLENGE\*\*\*" in the password field. Fetch will show you the server's challenge and prompt you for a response. See the [One-Time Password](#) help topic for more information.

6. *Does Fetch support resuming uploads and downloads?*

Fetch tries to detect stalls and network or server errors and automatically resume both uploads and downloads for you when they are interrupted. However, if you accidentally stop the transfer of a large file before it completes, yes, it is possible

to resume transfers for certain types of files. See the [resuming an interrupted download](#) or [resuming uploads](#) help topic for more information and limitations.

*7. Is Fetch Section 508 compliant?*

Fetch is partially compliant with the Section 508 standards. See our [compliance statement](#) webpage for more information.



Here are answers to commonly asked questions about passwords in Fetch.

1. [I don't remember my password, can you tell me what it is?](#)
2. [Why is Fetch rejecting my password when I know I entered it correctly?](#)
3. [How do I get the password out of a shortcut?](#)
4. [Can I change my password with Fetch?](#)
5. [Is my password being sent securely?](#)
6. [Why is Fetch asking me to enter my password?](#)
7. [Why is Fetch asking me to enter a passphrase?](#)

1. *I don't remember my password, can you tell me what it is?*

Please see [I don't know or I've forgotten my hostname, username, or password](#) in the [Solving Problems with Connecting](#) help topic.

2. *Why is Fetch rejecting my password when I know I entered it correctly?*

Please see [I received an error saying "login incorrect" or "rejected password"](#) in the [Solving Problems with Connecting](#) help topic.

3. *How do I get the password out of a shortcut?*

You can use the Keychain Access application to look at the password saved for a shortcut. When you create a shortcut and enter a password for it, the password is automatically saved in your [keychain](#).

To find your shortcut's password in the Keychain Access application:

- Open the Keychain Access application, which is in the Utilities folder inside your Applications folder.
- Look for an entry whose Name is the name of the server specified in the shortcut, and whose Kind is "Fetch password" (or possibly "Internet password").
- Display the entry's details by double-clicking on the entry.
- Click the "Show password" checkbox.
- When prompted, enter your keychain passphrase (usually the same as your Macintosh login or administrator password) to display the shortcut's password.

4. *Can I change my password with Fetch?*

No, that is not a feature of Fetch. Contact your service provider or network administrator for help changing your password.

5. *Is my password being sent securely?*

You can tell if your password is being sent securely by the padlock icon in the password prompt dialog. A locked padlock icon indicates your password will be protected by encryption. An unlocked padlock icon means your password will be sent insecurely, without encryption. There will also be a warning in the password dialog if your password will be sent insecurely.

Your password will be secure if you are using SFTP, FTP with TLS/SSL, or Kerberos (GSSAPI or KClient). If you are using plain FTP, your password is sent insecurely.

#### 6. *Why is Fetch asking me to enter my password?*

Sometimes Fetch will ask for your password outside of the [New Connection dialog](#). This can happen:

- Any time you open a connection without using the New Connection dialog (such as when you choose a [shortcut](#) or [recent connection](#) from the main menu, or use the Fetch widget or an Automator workflow to transfer files) and the password for that connection is not saved in your [keychain](#). If you check the **Add password to keychain** box after typing in your password to save it to your keychain, you will not be asked for the password again in the future.
- When you're opening a KClient ([Kerberos](#) v4) connection, even from the New Connection dialog. While Fetch uses your Kerberos tickets to encrypt your password, Kerberos v4 FTP servers require you to enter your password to access the server. GSSAPI connections, which use Kerberos v5, can forward your Kerberos ticket to the server to be used, so you are usually not asked for your password again when making GSSAPI connections.

#### 7. *Why is Fetch asking me to enter a passphrase?*

Fetch may ask you to enter a passphrase for an SSH key instead of, or in addition to, a password when you've set up public key authentication for an SFTP connection and your key is protected by a passphrase.



Here are answers to questions about licensing and registering Fetch. For general information about purchasing and licensing Fetch, see the [Licensing](#) help topic.

1. [Why does Fetch say that my serial number is not valid when I copied it from the receipt?](#)
2. [I purchased Fetch but lost my serial number; how can I get it?](#)
3. [Do registered users of Fetch 4 or Fetch 5 get a discount on upgrading to 5.6?](#)
4. [How should I distribute a licensed copy of Fetch to users at my educational or charitable organization?](#)
5. [Under what terms and conditions is Fetch licensed?](#)

1. *Why does Fetch say that my serial number is not valid when I copied it from the receipt?*

Each serial number is only valid with a certain name, the Serial Number Name, which is also included in the receipt. The Serial Number Name might be your company or organization name as opposed to your own name. Be sure you are entering the correct Serial Number Name along with the serial number. The Serial Number Name must be entered exactly as written in your receipt.

2. *I purchased Fetch but lost my serial number; how can I get it?*

Please fill out the [lost serial number form](#) on our website and we'll look in our records for your serial number.

3. *Do registered users of Fetch 4 or Fetch 5 get a discount on upgrading to 5.6?*

Upgrades to Fetch 5.6 are free for registered users of Fetch 5.5 and later, or if you purchased Fetch after January 28, 2009. Otherwise, registered users of any version of Fetch 5.3.1 or earlier can upgrade to Fetch 5.6 for the discounted price of \$10. To upgrade, choose [Purchase Upgrade](#) from the **Fetch** menu.

4. *How should I distribute a licensed copy of Fetch to users at my educational or charitable organization?*

We recommend that you add your institution's serial number into a copy of Fetch and distribute that copy to authorized users (e.g. by making it available for download from a server that is restricted to students and faculty, or by putting it on a CD-ROM that is only distributed to charity staff members). To add your serial number to a copy of Fetch 5.6 you should:

1. Copy Fetch 5.6 to a folder to which you have write permission.
2. Open Fetch 5.6 and dismiss the New Connection dialog if it appears.
3. Hold down Option, go to the **Fetch** menu, and choose **Enter Serial Number**.
4. Enter your institution's Fetch serial number name and serial number.
5. Quit Fetch.
6. [Optional] To confirm that the serial number was saved, control-click on Fetch and choose **Show Package Contents** from the Finder contextual menu. Then open the Contents folder, open the Resources folder, open the License.plist file, and verify that it contains your institution's serial number name and serial number.

5. *Under what terms and conditions is Fetch licensed?*

Fetch is licensed as-is; Fetch Softworks makes no warranty, express or implied, as to its suitability for any purpose. A single-user licensee may use Fetch on different computers (for example a laptop and a desktop), provided that Fetch is only used by the licensee on one computer at a time. A Fetch license for X users covers the simultaneous use of up to X copies of Fetch, and no more.



Here are answers to a few miscellaneous questions about Fetch.

1. [What is the relationship between Fetch Softworks and Dartmouth College?](#)
2. [What kind of dog is the Fetch mascot?](#)
3. [Who drew the Fetch mascot?](#)
4. [What does Regis Philbin have to do with Fetch?](#)

*1. What is the relationship between Fetch Softworks and Dartmouth College?*

Jim Matthews, the founder of Fetch Softworks, wrote Fetch as an employee of Dartmouth College. Fetch Softworks was created in order to license the Fetch source code and name from Dartmouth College, so that the product could be enhanced and marketed independently. Dartmouth College is not responsible for the statements and actions of Fetch Softworks.

*2. What kind of dog is the Fetch mascot?*

A mutt. And yes, it is a dog, not a koala.

*3. Who drew the Fetch mascot?*

Richard Nelson, an artist on Maui, who also happens to be the father-in-law of Fetch's author, drew the original 32x32 Fetch icon. The new 3D rendered mascot was commissioned by Tadashi Suzuki of Hart Computer, which sells a Japanese version of Fetch, and is used with permission.

*4. What does Regis Philbin have to do with Fetch?*

Directly, nothing. But winning money on Philbin's "Who Wants To Be A Millionaire" made it possible for Jim Matthews, Fetch's author, to license the software from Dartmouth College and create Fetch Softworks.



## [Fetch Help](#) > Troubleshooting & Support

Fetch works well for most people, most of the time; but that is little consolation if it isn't working for you! This part of the Fetch Help is the place to turn to when that happens. Please note that there are many thousands of Fetch users for every person providing technical support, so we are not equipped to provide extensive individual attention for each problem. Fortunately most problems are not one-of-a-kind, and the solutions may already be at hand.

That's why we recommend that you start first with the [Frequently Asked Questions \(FAQ\)](#), or by searching the Fetch Help by entering a term in the search box above.

Your next stop should be the [Fetch Softworks message boards](#), which let you benefit from the wisdom and experience of others, and vice versa. Fetch Softworks personnel will give top priority to addressing questions on the message boards.

Since we're always improving Fetch, you should check to see if you're using [the latest version](#), and if not, download it and see if it fixes the problem you're experiencing.

If you cannot find the answer via the above methods, you can submit your problem or question using the [Fetch Softworks feedback page](#). Or you can send email to [support@fetchsoftworks.com](mailto:support@fetchsoftworks.com), but it's preferable if you use the feedback page and provide the information it requests.

Note that many of the problems people experience with Fetch aren't really problems with Fetch itself. Publishing information on the Internet is still an involved process, and Fetch is only one part of the process. Fetch cannot fix HTML errors, teach your web server about new kinds of files, or change your password. Nor can it work if your network settings are misconfigured, or the actual network connection is unreliable. In some cases we will recommend that you try another FTP client, such as Interarchy, Transmit, or Vicomsoft FTP. We do this for two reasons: 1) we want to help you solve your problem, regardless of whether you end up using our software; and 2) if the problem persists with another program you can be fairly confident that neither Fetch nor the other program is at fault, and that gets you closer to finding out where the problem actually lies.



## [Fetch Help](#) > Reference

Select one of the topics below to get detailed information about Fetch.

If you don't see what you're looking for, check the [Using Fetch](#) list of topics, or use the Search field above.

[Menus](#)

[Toolbars](#)

[Preferences](#)

[Windows and Dialogs](#)

[Widget](#)

[Commands](#)

[Concepts](#)

[Messages and Warnings](#)

[Frequently Asked Questions](#)

[Terms You Should Know](#) / [Glossary](#)

The entire Fetch Help is available as a [downloadable PDF file](#) from the Fetch Softworks website. It serves as the Fetch manual.

 [Fetch Help](#) > [Reference](#) > **Menus**

The following topics provide detailed information about the Fetch menus and menu items.

[Fetch menu](#)

[File menu](#)

[Edit menu](#)

[View menu](#)

[Remote menu](#)

[Go menu](#)

[Shortcuts menu](#)

[Window menu](#)

[Help menu](#)

[Contextual menus](#)

## [Fetch Help](#) > [Menus](#) > **Fetch menu**

Select a command title for more information about that command, if available.

### [About Fetch](#)

Shows a window containing information about your copy of Fetch and quick ways to contact Fetch Softworks.

### [Check for Update](#)

Checks to see if you're using the latest version of Fetch, and provides an easy way to download updates if you're not.

### [Purchase](#) / [Purchase Upgrade](#)

Leads you through the purchasing process for a new version or upgrade of Fetch, and automatically enters your serial number after you've finished. Does not appear in the menu if you've already entered a serial number.

### [Enter Serial Number](#)

Lets you enter the name and serial number you have received to license Fetch. Disabled if you've already entered your serial number.

### [Install Dashboard Widget](#)

Installs a fresh, updated copy of the [Fetch widget](#).

### [Preferences](#)

Shows the Fetch preferences window.

### **Services**

The standard Mac OS X Services menu. Services are generally only available in text edit fields.

### **Hide Fetch**

Hides Fetch and all its windows.

### **Hide Others**

Hides all other applications and their windows.

### **Show All**

Show (unhide) all hidden applications.

### [Quit Fetch](#)

Quits Fetch. Any open connections will be closed, and any file transfers will be stopped (but you will be given an option to reconsider first).

## [Fetch Help](#) > [Menus](#) > **File menu**

Select a command title for more information about that command.

### [New Connection](#)

Opens the [New Connection dialog](#), where you can enter information to open a connection to a server.

### [Recent Connections](#)

Provides quick access to servers that you've recently visited.

### [Open](#)

Prompts you to open previously saved [shortcut documents](#), [shortcut list documents](#), or [mirror documents](#).

### [Close](#) / [Close All](#)

Closes the active window, or all windows when the Option key is pressed.

### [Save](#)

**Save** lets you save changes to a shortcut list window.

### [Save As](#) / [Save a Copy As](#)

**Save As** saves a shortcut list window to a new shortcut list document.

**Save a Copy As** saves the contents of a text window or media window to a local disk, or lets you save a copy of the Fetch Shortcuts window.

### [Save Droplet Shortcut](#) / [Save Shortcut](#)

**Save Droplet Shortcut** creates a [droplet shortcut](#) that you can drag and drop files to in the Finder to upload those files.

**Save Shortcut** creates a [shortcut document](#) that refers to an item selected in the file list.

### [Page Setup](#)

Shows the standard Page Setup dialog so you can set the paper size, orientation, and scaling for printed documents.

### [Print](#)

Shows the standard Print dialog and lets you print the Fetch Transcript window or any text window or media window.



Select a command title for more information about that command, if available.

### **Undo**

Undoes the last action. Most actions in Fetch **cannot** be undone, especially file transfer actions such as deleting, downloading, or uploading files. Not available when the last action cannot be undone.

### **Redo**

Redoes the last action that was undone, if possible. Most actions in Fetch **cannot** be redone.

### [Cut](#)

Cuts text or shortcuts to the clipboard.

### [Copy](#) / [Copy Fetch Address](#) / [Copy Web Address](#) / [Copy With Fetch Address](#)

Copies text, shortcuts, or URLs of items.

### [Paste](#)

Pastes text, Fetch and web address (URLs), shortcuts, or items to upload from the clipboard.

### [Delete](#)

Deletes shortcuts from shortcut list windows, removes entries from the Resume Download window, or deletes text in edit fields.

### **Select All**

Selects all the text in a window or field, all the files in a file list, or all the shortcuts in a shortcut list window.

### [Find](#) / [Find Again](#)

Searches for text in most Fetch windows.

### **Special Characters**

Displays the Mac OS character palette, where you can browse characters that may not be easily entered using the keyboard and insert them into text (such as a shortcut name).

 [Fetch Help](#) > [Menus](#) > **View menu**

Select a command title for more information about that command, if available.

### **Show/Hide Toolbar**

Shows or hides the toolbar, if the window has one.

### **Customize Toolbar**

Lets you add or remove items from a window's toolbar, if the window has one.

### **Stop**

Stops the current transfer or action. Only available when an action is in progress.

### **Refresh**

Updates the active transfer window's file list with the latest information from the server.

### **Set Font and Size**

Sets the font and size of text used by file lists in transfer windows, text windows, and shortcut list windows. Also sets the tab width used by text windows.



## [Fetch Help](#) > [Menus](#) > Remote menu

Select a command title for more information about that command.

### [Get](#) / [Get As](#) / [Get Other](#) / [Get Other As](#)

Downloads single or multiple files or folders from a server to your Macintosh.

### [Mode](#)

Lets you choose which mode (text, binary, or automatic) to use when getting or downloading files from a server.

### [Resume Download](#)

Lets you manually resume partially completed downloads.

### [Put](#)

Uploads single or multiple files or folders from your Macintosh to a server.

### [Format](#)

Lets you choose which format to use when putting or uploading files to a server.

### [New Folder](#)

Makes a new folder on the server.

### [Mirror](#)

Automatically copies all the new or changed files from a Macintosh folder to a server folder, or vice versa.

### [Quick Look](#) / [Close Quick Look](#)

Lets you display files on a server without leaving Fetch. Many different kind of files can be displayed, including text files, images, sounds, movies, PDFs, and Microsoft Word and Excel files. (Only available on Mac OS X 10.5 and later.)

### [View as Text](#)

Lets you view any file on a server as plain text without leaving Fetch. (Only available on Mac OS X 10.5 and later.)

### [View](#) / [View As](#) / [View Other](#)

Lets you view text files or media files (images, sounds, movies) on a server without leaving Fetch. (Only available on Mac OS X 10.4.)

### [Edit](#) / [Edit With](#) / [Edit Other](#) / [Edit New Text File](#)

Opens files for editing in an editor application. Changes will be saved back to the server automatically.

### [Edit With submenu](#)

Lets you temporarily use a different editor for editing the selected files. Changes will be saved back to the server automatically.

### [Get Info](#)

Displays a window containing detailed information about the selected remote files or folders, and lets you rename or set permissions on the items.

### [WebView](#)

Displays the currently selected items in your web browser (or set up WebView for the current server, if necessary).

### [Delete Item](#) / [Delete Other Item](#)

Deletes files and folders on a server.

### [View Verbose File List](#)

Lets you view an extended file list for the current folder in a separate text window.

### [Send FTP Commands](#)

Sends arbitrary FTP commands to an FTP server, with special support for searching (SITE INDEX) and setting upload permissions.

### [Enable Encryption](#) / [Disable Encryption](#)

Turns encryption on or off during an FTP session that is using FTP with TLS/SSL or Kerberos (GSSPAI or KClient) security.



[Fetch Help](#) > [Menus](#) > **Go menu**

Select a command title for more information about that command, if available.

### [Back](#)

Returns you to the remote folder you were previously viewing.

### [Open Folder](#)

Makes the selected folder the current folder and displays its contents in the file list.

### [Parent Folder](#)

Takes you to the parent folder (enclosing folder) of the folder you're currently viewing.

### [Home](#)

Takes you to your home folder on a server.

### [Go to Folder](#)

Lets you change to another folder on the server by typing its path.

### [Recently visited folders](#) [list of folders]

Choosing a folder takes you to that folder. Fetch remembers the folders you've recently visited on each server, and lists them in the Go menu so you can return to them quickly.

### [Clear Menu](#)

Clears out the list of recently visited folders shown in the Go menu.



## [Fetch Help](#) > [Menus](#) > **Shortcuts menu**

Select a command title for more information about that command, if available.

### [Show Shortcuts](#)

Opens or brings to the front the [Fetch Shortcuts](#) window, which lists the shortcuts you've created to files, folders, and servers.

### [New Shortcut](#)

Lets you create a new shortcut to access remote files or folders quickly.

### [Edit Shortcut](#)

Lets you modify an existing shortcut.

### [New Shortcut List](#)

Opens a new [shortcut list window](#), where you can create a list of shortcuts separate from the main [Fetch Shortcuts](#) window.

### [List of shortcuts](#)

At the bottom of the **Shortcuts** menu is the list of shortcuts from the [Fetch Shortcuts](#) window, and also a list of [Bonjour](#) servers. Choosing a shortcut opens a connection. To add a shortcut to this list, use the **New Shortcut** command or see the [creating a shortcut](#) help topic. To change or remove a shortcut, see the [deleting or changing a shortcut](#) help topic.

## [Fetch Help](#) > [Menus](#) > **Window menu**

Select a command title for more information about that command, if available.

### **Minimize Window / Minimize All Windows**

Minimizes a window, putting it in the Dock. To open the window again, click its icon in the Dock. Hold down the Option key and choose this command to minimize all Fetch windows.

### **Zoom**

Enlarges a window to show as much information as possible. Choose Zoom again to return the window to its previous size.

### **Bring All to Front / Arrange in Front**

Brings all the windows belonging to Fetch to be in front of the windows of other applications' windows. Hold down the Option key and choose this command to bring all Fetch windows to the front and tile them neatly.

### **[Fetch Shortcuts](#)**

Opens or brings to the front the [Fetch Shortcuts window](#), which lists the shortcuts you've created to files, folders, and servers.

### **[Fetch Transcript](#)**

Opens or brings to the front the [Fetch Transcript window](#), which shows a complete record of all the commands sent and responses received from servers.

### **Currently Open Windows [list of window titles]**

At the bottom of the **Window** menu is a list of the windows currently open in Fetch. Choosing a window's name makes it the active window. The active window has a checkmark next to its name. Any minimized windows have diamonds (◆) next to their name.

 [Fetch Help](#) > [Menus](#) > **Help menu**

The Help menu contains the following commands:

**Fetch Help**

Displays the Fetch Help. You are currently viewing the Fetch Help.

**(Window Name) Help**

Shows help specific to the active window or dialog.



## [Fetch Help](#) > [Menus](#) > Contextual menus

Contextual menus in Fetch provide convenient access to often-used commands via a context-sensitive menu. For example, if you select a file in a Fetch transfer window, the contextual menu contains various commands that can be performed on that file: **Get**, **Quick Look**, **Edit**, **Delete**, **Copy Fetch Address**, etc.

To display the contextual menu, press the Control key and click an item; contextual menus are available in Fetch [transfer window](#) and [shortcut list windows](#), including the [Fetch Shortcuts window](#).

For further information about each contextual menu, see:

- [File list contextual menu](#) help topic
- [Shortcut list window contextual menu](#) help topic

There are two commands that are only available through contextual menus. The [Clone Window](#) command lets you create a second connection to a server by cloning a current connection. The [Duplicate](#) command lets you duplicate a shortcut in a shortcut list, or duplicate a file on a server. Duplicating remote files is done as a [server-to-server transfer](#).



## [Fetch Help](#) > [Menus](#) > **File List contextual menu**

When you Control-click in a file list, the file list contextual menu appears. The contents of the menu, and the behavior of the commands, depend on whether you've clicked on a file, a folder, or no items.

Select a command title for more information about that command.

### **Transfer Window Help**

Displays the help for transfer windows.

### [Copy Fetch Address](#) / [Copy Web Address](#)

Copies a [Uniform Resource Locator \(URL\)](#) referring to the selected items to the clipboard. When no items are selected, copies a URL to the current folder instead of an item in that folder.

### [Paste](#)

Pastes items to upload from the clipboard.

### [Duplicate](#)

Duplicates the selected files on a server via a server-to-server transfer. Only available when files are selected.

### [Clone Window](#)

Creates a second connection to a server by duplicating the current transfer window.

### [Refresh](#)

Updates the file list with the latest information from the server.

### [New Shortcut](#)

Lets you create a new shortcut to access the selected files or folders quickly, or when no items are selected, to create a shortcut to the current folder.

### [Get](#) / [Get As](#)

Downloads the selected files and folders from the server to your Macintosh.

### [Open Folder](#)

Makes the selected folder the current folder and displays its contents in the file list. Only available when a single folder is selected.

### [New Folder](#)

Creates a new folder in the current folder.

### [Quick Look](#) / [Close Quick Look](#)

Display the selected files without leaving Fetch. Many different kind of files can be displayed, including text files, images, sounds, movies, PDFs, and Microsoft Word and Excel files. Only available when files are selected. (Only available on Mac OS X 10.5 and later.)

### [View as Text](#)

View the selected files as plain text without leaving Fetch. Only available when files are selected. (Only available on Mac OS X 10.5 and later.)

### [View](#) / [View As](#)

View the selected text files or media files (images, sounds, movies) without leaving Fetch. Only available when files are selected. (Only available on Mac OS X 10.4.)

### [Edit](#) / [Edit With](#)

Opens the selected files for editing in an editor application. Changes will be saved back to the server automatically. Only available when files are selected.

### [Edit With submenu](#)

Lets you temporarily use a different editor for editing the selected files. Changes will be saved back to the server automatically. Only available when files are selected.

### [Get Info](#)

Displays an [info window](#) containing detailed information about the selected remote files or folders, and lets you rename or set permissions on the items. Only available when items are selected.

### [WebView](#)

Displays the currently selected items in your web browser (or set up WebView for the current server, if necessary).

### [Delete Item](#)

Deletes the selected files and folders from the server. Only available when items are selected.



## [Fetch Help](#) > [Menus](#) > **Shortcut List Window contextual menu**

When you Control-click in the [Fetch Shortcuts](#) window or another [shortcut list window](#), the shortcut list window contextual menu appears. The contents of the menu, and the behavior of the commands, depend on what type of shortcut you've clicked on. Most commands will not be available unless you Control-click a shortcut.

Select a command title for more information about that command.

### **Shortcut List Help**

Displays the help for shortcut list windows.

### **Cut**

Cuts the selected shortcuts to the clipboard.

### **Copy**

Copies the selected shortcuts to the clipboard.

### **Paste**

Pastes Fetch or web addresses (URLs) or shortcuts from the clipboard.

### **Delete**

Deletes the selected shortcuts.

### **Duplicate**

Duplicates the selected shortcuts.

### **New Shortcut**

Displays the [New Shortcut dialog](#) so you can create a new shortcut.

### **Edit Shortcut**

Displays the [Edit Shortcut dialog](#) so you can edit the selected shortcut. Only available when a single shortcut is selected.

### **Get / Get As**

Downloads the files and folders referenced by the selected shortcut from the server to your Macintosh. A connection to the server containing the items will be opened if necessary.

### **Open**

Opens the folders referenced by the selected shortcuts in transfer windows. Connections to the servers containing the folders will be opened if necessary. Only available when folder shortcuts are selected.

### **Put**

Uploads single or multiple files or folders from your Macintosh to the server referenced by the selected shortcut. Only available when a single folder shortcut is selected.

### **Quick Look / **Close Quick Look****

Display the files referenced by the selected shortcuts without leaving Fetch. Many different kind of files can be displayed, including text files, images, sounds, movies, PDFs, and Microsoft Word and Excel files. Only available when a single file shortcut is selected. (Only available on Mac OS X 10.5 and later.)

### [View as Text](#)

Displays the files referenced by the selected shortcuts as text files without leaving Fetch. Connections to the servers containing the files will be opened if necessary. Only available when file shortcuts are selected. (Only available on Mac OS X 10.5 and later.)

### [View](#) / [View As](#)

Displays the files referenced by the selected shortcuts as text files or media files without leaving Fetch. Connections to the servers containing the files will be opened if necessary. Only available when file shortcuts are selected. (Only available on Mac OS X 10.4.)

### [Edit](#) / [Edit With](#)

Opens the files referenced by the selected shortcuts for editing in an editor application. Changes will be saved back to the server automatically. Connections to the servers containing the files will be opened if necessary. Only available when file shortcuts are selected.

### [Edit With submenu](#)

Lets you temporarily use a different editor for editing the files referenced by the selected shortcuts. Changes will be saved back to the server automatically. Only available when file shortcuts are selected.

### [WebView](#)

Displays the currently selected items in your web browser (or set up WebView for the current server, if necessary).

## [Fetch Help](#) > Transfer Window Toolbar

The toolbar at the top of transfer windows provides easy access to frequently used commands.



Fetch ships with a default set of buttons in the toolbar, but you can add other buttons, remove any of the default ones, or return the toolbar to its original state. See below for a complete list of the buttons available and a short description of what they do.

To customize the toolbar:

Choose **View > Customize Toolbar**, or Control-click in the toolbar and choose **Customize Toolbar** from the contextual menu.

You can also hide the toolbar, or change it to show smaller icons or only icons or only text.

To hide the toolbar:

Choose **View > Hide Toolbar**. To show it again, choose **View > Show Toolbar**. You can also click the toolbar button in the upper-right corner of the transfer window.

To change the appearance of the toolbar:

Control-click in the toolbar and choose one of the following options:

- **Icon & Text** – Show both icons and names in the toolbar
- **Icon Only** – Show toolbar buttons as icons without names
- **Text Only** – Show toolbar buttons as only names
- **Use Small Size** – Display the icons and names at a smaller size

The default buttons in the toolbar are listed below. Select a button title for more information.



Go back to the remote folder you were viewing before the current one.



Click this button to display a list of all the folders that contain the current one. Choose a folder to go to it.



Click this button to display a list of folders you've recently visited on

the current server. Choose a folder to go to it.



### Get

Download one or more files or folders from a server to your Macintosh.



### Put

Upload one or more files or folders from your Macintosh to a server.



### Quick Look

Display files on a server without leaving Fetch. Many different kind of files can be displayed, including text files, images, sounds, movies, PDFs, and Microsoft Word and Excel files. (Only available on Mac OS X 10.5 and later.)



### View

View text files or media files (images, sounds, movies) on a server without leaving Fetch. Fetch will guess which kind of file you have selected, but if it guesses incorrectly, hold down the Option key to be asked what kind of file you are viewing. (Only available on Mac OS X 10.4.)



### Edit

Edit files in an editor application and save changes back to the server automatically.



### Get Info

Display a window containing detailed information about the selected remote files or folders, and lets you rename or set permissions on the items.



### WebView

Display the currently selected items in your web browser (or set up WebView for the current server, if necessary). The WebView button's icon will always be the icon of your default web browser, so the icon may be different if you have set a browser other than Safari as your default web browser.



### New Folder

Make a new folder on the server.



### Delete

Delete files or folders on a server.

The optional items that you can add to the toolbar are listed below. Select a button title for more information (if available).



### [Format](#)

Choose which format to use when putting or uploading files to the server.



### [Go to Folder](#)

Change to another folder on the server by typing its name or path.



### [Home](#)

Take you to your home folder on the server.



### [Mirror](#)

Show the Mirror window, where you can copy all the new or changed files from a Macintosh folder to a server folder, or vice versa.



### [Mode](#)

Choose what mode (automatic, text, or binary) to use when getting or downloading files from a server.



### [New Shortcut](#)

Create a new shortcut to access remote files or folders quickly.



### [Parent](#)

Change the current folder to be the parent folder of the folder you're currently viewing, that is, go up one level. You can drag items to the **Parent** button to move them to the parent folder.



### [Resume Download](#)

Resume partially completed downloads.



### [Send FTP Commands](#)

Send arbitrary FTP commands to an FTP server, with special support for searching (SITE INDEX) and setting upload permissions.



### [View as Text](#)

View files on a server as plain text without leaving Fetch. (Only available on Mac OS X 10.5 and later.)



### **Separator**

Insert a vertical line in the toolbar so you can group items together.



### **Flexible Space**

Insert blank space in the toolbar between items.



### **Customize**

Display the Customize Toolbar dialog.

The Preferences allow you to change and control aspects of Fetch's behavior. The Fetch Preferences window is divided into several panes, each one containing preferences relating to a different area of Fetch. Click the tabs to switch among the panes.

Select a pane title below for information about the preferences in that pane.

If you don't see what you're looking for, use the Search field above.

### [General](#)

Preferences in the General Preferences pane let you modify general aspects of Fetch's behavior, such as notification, default shortcut, and server message options.

### [Download](#)

Preferences in the Download Preferences pane affect how and where Fetch downloads files.

### [Upload](#)

Preferences in the Upload Preferences pane affect the format and names of files uploaded with Fetch.

### [WebView](#)

The WebView Preferences pane lists the WebView entries you have created, and lets you add, delete, or edit them.

### [Proxy](#)

Preferences in the Proxy Preferences pane allow Fetch to work with your organization's proxy server or firewall.

### [Security](#)

Preferences in the Security Preferences pane control warnings and options for SFTP, FTP with TLS/SSL, Kerberos, and One-Time Password connections.

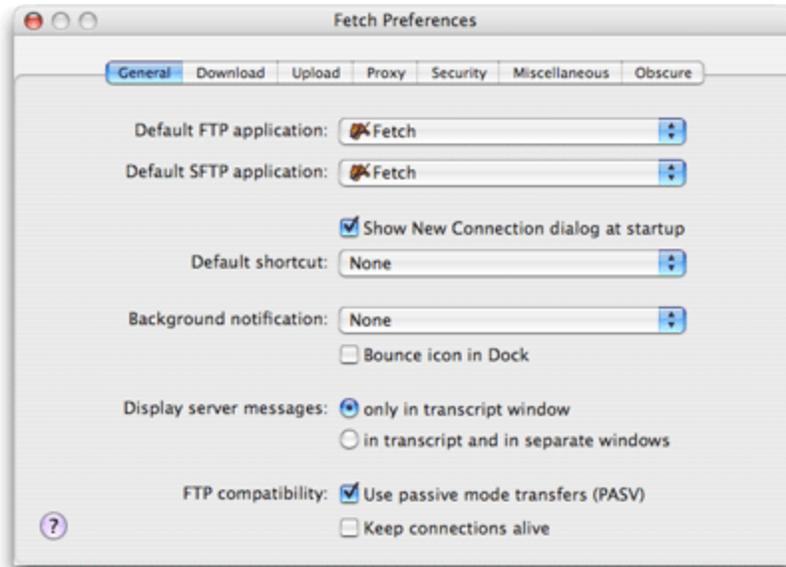
### [Miscellaneous](#)

Preferences in the Miscellaneous Preferences pane include file list display options, character encoding, and duplicate connections.

### [Obscure](#)

Preferences in the Obscure Preferences pane cover unusual situations that most users will not encounter.

Preferences in the General Preferences pane let you modify general aspects of Fetch's behavior, such as notification, default shortcut, and server message options.



### Default FTP application

Choose which application will be opened when you click on an FTP or FTP with TLS/SSL (FTPS) link or [URL](#). If the application you want isn't listed, choose **Other** and select the application.

### Default SFTP application

Choose which application will be opened when you click on an SFTP link or [URL](#). If the application you want isn't listed, choose **Other** and select the application.

### Show New Connection dialog at startup

When this box is checked, Fetch will automatically show the [New Connection dialog](#) when it is started. Checked by default.

### Default shortcut

You can automatically fill out the [New Connection dialog](#) with information from a shortcut by choosing a shortcut from this pop-up menu. If you do not want the New Connection dialog to be automatically filled out, choose the **None** item. Only shortcuts listed in the [Fetch Shortcuts](#) window are listed here. If the shortcut you want is not listed, you may need to [create a shortcut](#) first. **None** is the default setting.

### Background notification

Choose the sound (if any) that you would like Fetch to make when it completes a task while in the background. Choose **None** for no sound, **Alert Sound** to play the sound that's set as the alert sound in the Mac OS Sound preferences pane, **Speak** if you want Fetch to say "Fetch task complete," **Bark** to hear Fetch bark, or choose one of the other sounds from the list. **None** is the default setting.

### Bounce icon in Dock

Check this box if you want Fetch to bounce its icon in the Dock whenever it finishes a task in the background. Unchecked by default.

### Display server messages

Some servers display special information when the connection is established, such as hours of operation, use restrictions, number of open connections, and information about the files on the server.

You can set this preference to either **only in transcript window** — to display server messages only in the [Fetch Transcript](#) window — or **in transcript and in separate windows** — to display server messages in both separate [message windows](#) and the Fetch Transcript window. **Only in transcript window** is the default setting.

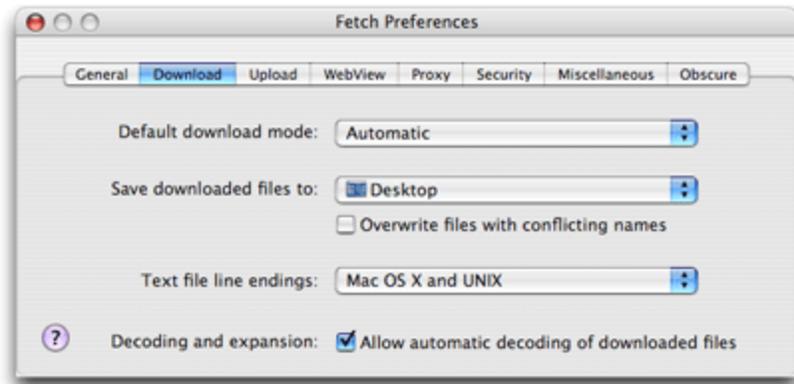
### Use passive mode transfers (PASV)

When this box is checked, Fetch will initiate data connections, rather than asking servers to connect back to your Macintosh. Some routers, firewalls, and NAT devices prevent incoming connections; checking this box makes it possible to use Fetch behind those devices. While Fetch's Automatic Passive Mode feature usually correctly determines when to use passive or active modes, you should leave this option checked unless you have a good reason not to. See the [Active and passive FTP](#) help topic for more information. Checked by default.

### Keep connections active

When this box is checked, Fetch will periodically send commands to the server to keep the connection open. Otherwise, Fetch will drop a connection if it has not been used for five minutes (the connection will be automatically restored when needed). Note that it is very impolite to keep a connection open to a public-access FTP server for a long period; many computers limit the total number of connections, and your connection may prevent others from using the server. Unchecked by default.

Preferences in the Download Preferences pane affect how and where Fetch downloads files.



### Default download mode

This pop-up menu specifies the default download mode for new [transfer windows](#). The options are explained further in the [download modes](#) help topic. You should choose **Automatic** unless you have a specific reason not to.

You can temporarily override this setting using the **Mode** submenu in [Remote menu](#).

**Automatic** is the default setting.

### Save downloaded files to

This pop-up menu specifies which folder Fetch will save items to when you download them. If the folder you want isn't listed, choose the **Other** item and select the desired folder. You can also drag a folder's icon from the Finder to the pop-up menu to set it as the download folder. Choose **Ask every time** to have Fetch prompt you for a location to save the downloaded files whenever you use the [Get](#) command. The Downloads folder is the default setting on Mac OS X 10.5 and later; the Desktop folder is the default setting on Mac OS X 10.4.

### Overwrite files with conflicting names

When this box is checked, Fetch will overwrite an existing Mac file if one with the same name is in the same folder as a file being downloaded. When this box is unchecked, Fetch will add a number to the filename of the newly downloaded file to distinguish it.

If you download a folder with the same name as an existing subfolder in the download folder, instead of overwriting or creating a new folder, Fetch will put the files from the remote folder inside the local folder of the same name.

Unchecked by default.

### Text file line endings

This pop-up menu lets you choose which style of line ending Fetch should use for downloaded text files. Choose **Mac OS X and UNIX** to use a single linefeed (LF). Choose **Mac OS (9 and older)** to use a single carriage return (CR) — some older Macintosh programs may still expect that style of line ending. Choose **Windows**

**and DOS** to use a carriage return followed by a linefeed (CRLF). **Mac OS X and UNIX** is the default setting.

### **Allow automatic decoding of downloaded files**

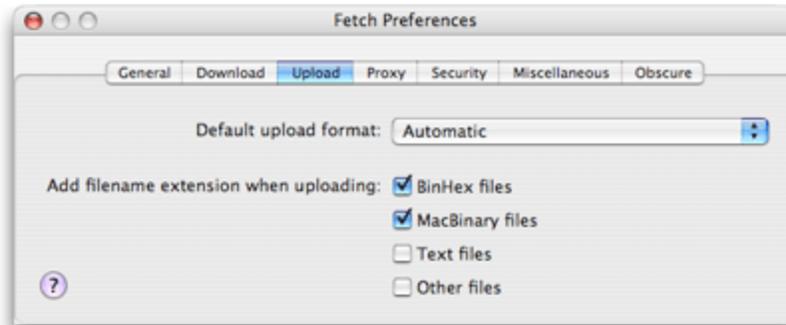
When this box is checked, Fetch will try to decode or expand files automatically on download, as specified by that kind of file's **Automatically decode files like this checkbox** in the [Transfer Options](#) pane of the [info window](#).

When this box is unchecked, downloaded files will never be automatically decoded or expanded by Fetch.

Note that unchecking this box will disable Fetch's automatic decoding of BinHex, MacBinary, and AppleSingle files in addition to disabling Zip and StuffIt archive expanding.

Checked by default.

Preferences in the Upload Preferences affect the format and names of files uploaded with Fetch.



### Default upload format

This pop-up menu lets you choose the format to use when uploading files. Usually you should use **Automatic**, where Fetch will try to determine the right format to use by examining the file in question. You can learn more about the other choices in the [upload formats](#) help topic. The [archive formats](#) cannot be set as your default upload format.

You can temporarily override this setting using the **Format** submenu in the [Remote menu](#). You can also temporarily override it in the choose file prompt when you use the [Put](#) command.

**Automatic** is the default setting.

### Add filename extension when uploading: BinHex files

If this box is checked, Fetch will add an ".hqx" extension to the names of files uploaded in [BinHex](#) format. The ".hqx" extension will be added in addition to any existing extension a file might have. Using this common extension makes it more likely that the files will be retrieved intact when downloaded again. Checked by default.

### Add filename extension when uploading: MacBinary files

If this box is checked, Fetch will add a ".bin" extension to the names of files being uploaded in [MacBinary](#) format. The ".bin" extension will be added in addition to any existing extension a file might have. Using this common extension makes it more likely that the files will be retrieved intact when downloaded again. Checked by default.

### Add filename extension when uploading: Text files

If this box is checked, Fetch will add an appropriate extension to the names of files being uploaded in [Text](#) format (for example, ".txt" for text files, ".html" for HTML files, ".c" for C files, and so forth), if the files don't already have an extension. Using these common extensions makes it more likely that text files will be retrieved intact when downloaded again. Unchecked by default.

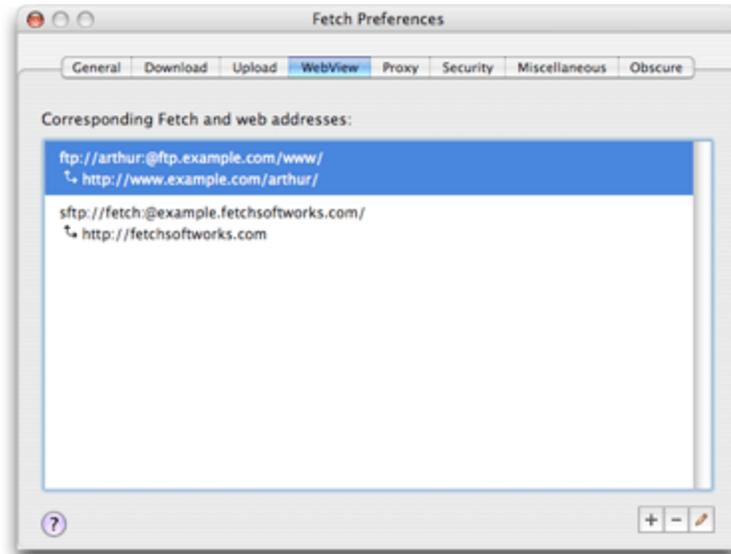
### Add filename extension when uploading: Other files

If this box is checked, Fetch will add an appropriate extension to the names of

files being uploaded in [Binary \(Raw Data\)](#) format, if the files don't already have an extension. For example, ".jpg" will be added to JPEG files, ".doc" to Microsoft Word files, and so forth. Many systems determine the type of a file from its name extension; for example, a JPEG file is handled better when its name ends in ".jpg". Unchecked by default.

## [Fetch Help](#) > [Preferences](#) > **WebView Preferences**

The WebView Preferences pane lists the [WebView](#) entries you have created, and lets you add, delete, or edit them.



Each WebView entry specifies a correspondence between a Fetch address and a web address. Usually there will be one entry for each server you use. For each entry, the Fetch address is listed first (usually starting with ftp, ftps, or sftp), followed by its corresponding web address (usually starting with http or https). The addresses are displayed as [URLs \(Uniform Resource Locators\)](#).

To add a new WebView entry, click the Add (+) button. The [WebView dialog](#) will be displayed, prompting you to enter or choose a Fetch address and web address.

An easier way to create WebView entries is to connect to your file server, and then click the **WebView** button in the toolbar. Fetch will display a version of the WebView dialog that prompts you for just the web address, since it can get the Fetch address of your server from the connection.

To delete a WebView entry, select an entry and click the Delete (-) button, or press the Delete key. **WARNING:** Deleting an entry is not undoable.

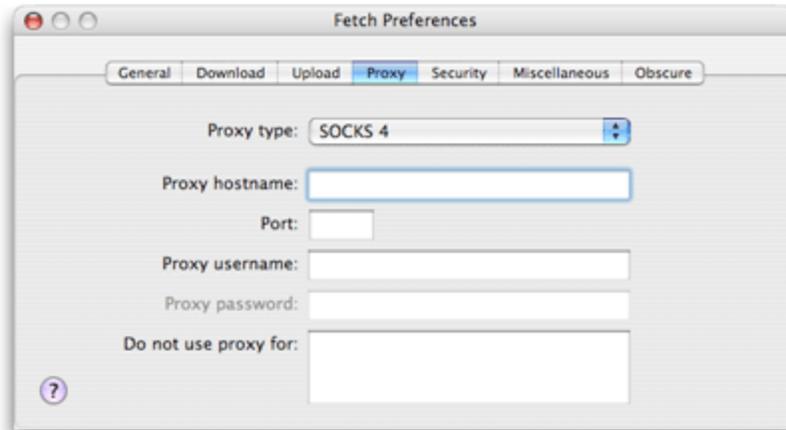
To change or edit a WebView entry, select an entry and click the Edit (✎) button or press the Return key, or double-click on an entry. The [WebView dialog](#) will be displayed, and you can change the Fetch address or the web address.

See the [WebView](#) topic for more information about WebView.

Related topics:

- [WebView](#)
- [Setting up WebView](#)
- [WebView dialog \(with Fetch address\)](#)
- [WebView command](#)
- [Copy Web Address](#)

Preferences in the Proxy Preferences pane allow Fetch to work with with proxy servers or firewalls that some sites use for security reasons.



Some networks only allow FTP sessions from a special computer called a proxy server. In that case, Fetch needs to connect to the proxy server, which in turn connects to the actual server. These preferences allow you to specify the information Fetch needs to work with a proxy server, if your network uses one.

You should consult with your network administrator or help desk before changing these preferences. They should be able to tell you what you need to enter, if anything. Having a proxy server is relatively uncommon, so you should leave these settings blank unless you have specifically been told otherwise.

With the exception of SOCKS proxies, the proxy preferences only apply to FTP connections, they do not affect SFTP connections. If you use a SOCKS proxy, the proxy preferences apply to both FTP and SFTP connections.

Fetch does not support FTP or SFTP through HTTP proxy servers.

These preferences supersede the proxy settings in the **Network** pane of **System Preferences**. Fetch does not use the System Preferences proxy settings.

### **Proxy type**

Choose the type of proxy server used by your network from this pop-up menu. Your network administrator should be able to tell you what to choose. Choose **None** if your site does not have a firewall or proxy server.

### **Proxy hostname**

Consult with your firewall or network administrator for the proper information to put in this field. Leave blank if your site does not have a firewall or proxy server.

### **Port**

Consult with your firewall or network administrator for the proper information to put in this field. Leave blank if your site does not have a firewall or proxy server.

### **Proxy username**

Consult with your firewall or network administrator for the proper information to put in this field. Leave blank if your site does not have a firewall or proxy server.

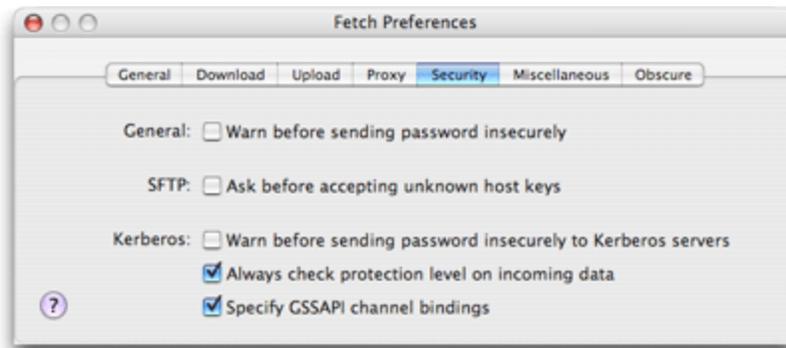
### **Proxy password**

Consult with your firewall or network administrator for the proper information to put in this field. Leave blank if your site does not have a firewall or proxy server.

### **Do not use proxy for**

Proxy servers and SOCKS gateways may not accept connections bound for local servers (servers that are inside your network firewall). If that is the case on your network, you can enter the name of your local domain in this field, and Fetch will make direct connections to servers in that domain. So, if you entered "example.com", Fetch would not use the proxy server or SOCKS gateway for connections to "ftp.example.com", "internal.example.com", or "example.com", but it would continue to use the proxy server or gateway for servers that are not in the "example.com" domain. You can enter multiple domain names (separated by commas, spaces, or on separate lines) if there are multiple domains inside your firewall (for example, your firewall might protect "research.example.com" and "sales.example.com" while "ftp.example.com" is outside the firewall; in that case, you would enter "research.example.com, sales.example.com"). Your network administrator should be able to tell you if you need to enter anything in this field. Leave blank if your site does not have a firewall or proxy server.

Preferences in the Security Preferences pane control warnings and options for [SFTP](#), [Kerberos](#), and [One-Time Password](#) connections. Also see the [Security](#) help topic for more information.



### **General: Warn before sending password insecurely**

When this box is checked, Fetch will warn you whenever you try to send your password unencrypted and insecurely. This is good if you usually connect to secure servers via SFTP, FTP with TLS/SSL, Kerberos, or One-Time Password, because you will be warned if you accidentally try to connect to an insecure server, or if a secure server isn't working properly and requests a password insecurely. However, when this preference is selected, you will receive warnings even if a server does not support secure connections — and most FTP servers require unencrypted passwords. Unchecked by default.

### **SFTP: Ask before accepting unknown host keys**

When this box is checked, Fetch will warn you when you try to connect to an SFTP server you haven't connected to previously. Accepting an SSH host key from a server says, "I trust this server from now on." If you want to double-check that you really want to trust a server before connecting, check this preference. See the [Host authenticity](#) help topic for more information. Unchecked by default.

### **Kerberos: Warn before sending password insecurely to Kerberos servers**

When this box is checked, Fetch will warn you when you try to send your password insecurely to a server that supports Kerberos, that is, you are not taking advantage of the server's security support. This could happen because you didn't specify **FTP with KClient** or **FTP with GSSAPI** in the New Connection dialog when connecting to the server, because the server is not working properly, or possibly because the server mistakenly claims to support security. This preference does not apply to SFTP or One-Time Password connections. See [this help topic](#) for more information. Unchecked by default.

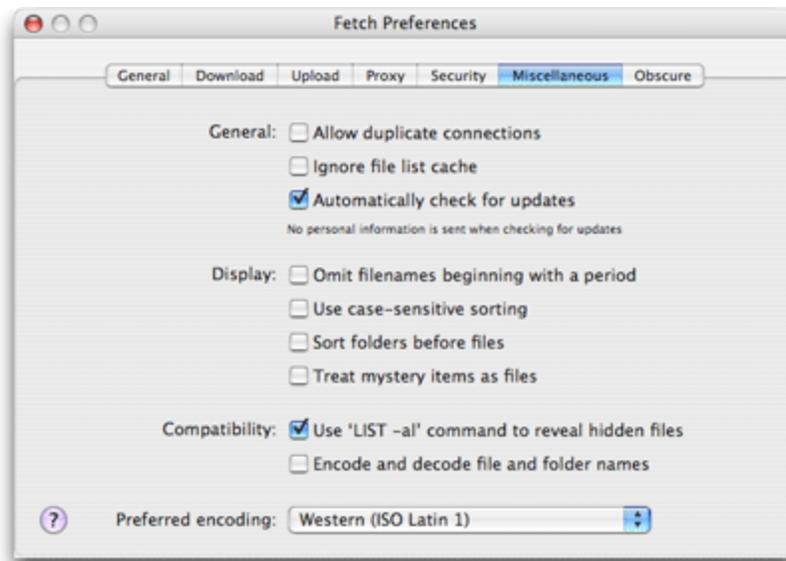
### **Kerberos: Always check protection level on incoming data**

When this box is checked, Fetch will warn you when a server starts sending unencrypted data over an encrypted Kerberos connection. If that happens, it could mean that the server, or your connection to it, has been compromised. This preference does not apply to SFTP or FTP with TLS/SSL connections. See [this help topic](#) for more information. Checked by default.

### **Kerberos: Specify GSSAPI channel bindings**

When this box is checked, Fetch will specify GSSAPI channel bindings. If you are connecting using **FTP with GSSAPI** from behind a Network Address Translation (NAT) device or router, you may need to uncheck this box to get a successful connection. Consult your network administrator or support if you continue to have problems. Checked by default.

Preferences in the Miscellaneous Preferences pane cover various aspects of Fetch, such as display options for the file list, character encoding, and whether to allow duplicate connections.



### Allow duplicate connections

When this box is checked, Fetch will let you perform actions such as **Get** or **Go to Folder** when a transfer window is already busy. In that case, Fetch will open a new, duplicate connection to perform the action. This can be convenient, but it is easy to accidentally create new connections when you did not realize that the original connection was busy.

When this box is unchecked, Fetch will not allow you to perform actions that would result in a duplicate connection when a transfer window is busy.

Unchecked by default.

### Ignore file list cache

When this box is checked, Fetch will always get a fresh file list when it enters a remote folder, instead of possibly using a stored copy in its cache. Otherwise Fetch will try to save time by using a cached file list. When Fetch uses a cached file list, it displays above the file list the date when the list was downloaded (for example, "As of 10/15/2004 12:15 PM"). For more information, see the [file list](#) help topic. Unchecked by default.

### Automatically check for updates

When this box is checked, Fetch will periodically look to see if a newer version of Fetch is available. This check only happens when you have a network connection. If an update is available, Fetch will display a dialog offering you an opportunity to download and install the update. No personal information is sent or disclosed by the update checks. See the [Software Update Privacy](#) help topic for more information. You can also check for updates manually by choosing the [Check for Update](#) command from the **Fetch** menu. Checked by default.

## Omit filenames beginning with a period

When this box is checked, Fetch will not display items whose names begin with a period ("dot files") in file lists. This can reduce the clutter when connecting to UNIX servers. When this box is unchecked, all items will be shown. Unchecked by default.

## Use case-sensitive sorting

When this box is checked, Fetch will consider case when sorting item names in file lists or shortcut names in shortcut list windows, so that all the names starting with upper case letters will be listed before ones starting with lower case. This is similar to the default behavior of the UNIX `ls` command. When this box is unchecked, Fetch will not take case into account when sorting item or shortcut names. Unchecked by default.

## Sort folders before files

When this box is checked, Fetch will always list folders first when you sort file lists by name. When this box is unchecked, items will be sorted alphabetically regardless of their type. Unchecked by default.

## Treat mystery items as files

On some servers, Fetch cannot immediately distinguish between files from folders. To successfully transfer or delete an item, Fetch needs to know whether it is a file or folder. Normally Fetch will start a transfer by first treating a mystery item as a folder, and trying to enter that potential folder. If entering the folder succeeds, Fetch knows the item is a folder. If Fetch gets an error from the server saying the item is not a folder, Fetch knows the item is a file. Unfortunately, a small number of servers (including MVS and Tandem servers) do not return an error when Fetch tries to enter a "folder" that is really a file, so Fetch erroneously concludes that it is getting or deleting a folder, and the action fails.

When this box is checked, Fetch will instead begin by assuming that mystery items are files, and will not try treating them as folders, so that transfers may work successfully on servers that do not return errors correctly. When this box is unchecked, Fetch will attempt to enter the potential folder first, as described above.

Unchecked by default.

## Use 'LIST -al' command to reveal hidden files

Some UNIX and Windows servers fail to show "dot files" (files and folders whose names begin with periods) in file lists, even when you have the **Omit filenames beginning with a period** preference unchecked, because of the way those servers implement the LIST FTP command. This preference tells Fetch to send a non-standard command to those servers, telling them to reveal the dot files. Usually the non-standard command does not cause any problems with servers that do implement the LIST command correctly, however, if your server has problems when getting file lists, uncheck this preference. Checked by default.

## Encode and decode file and folder names

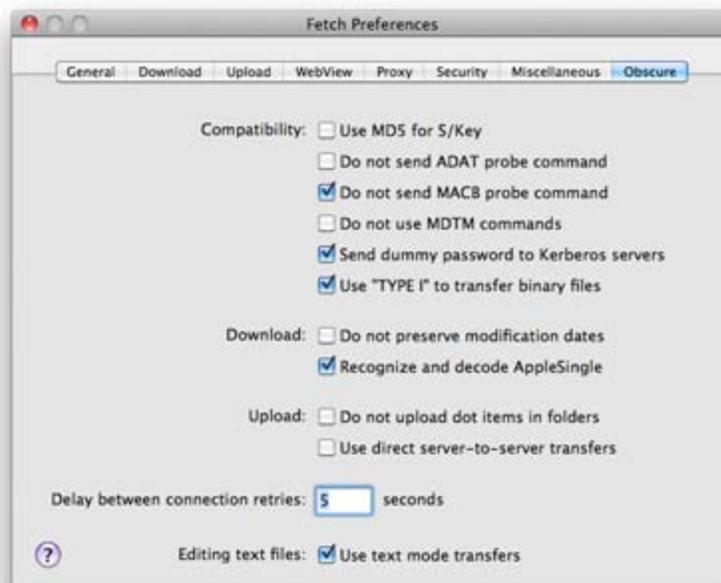
If this box is checked, Fetch will preserve special characters in item names by performing a special conversion of the names of local files and folders when putting them on the server, and vice versa. Fetch will encode and decode the

names according to the AppleSingle 7–Bit ASCII standard: slash (/), percent (%), null, and all 8–bit extended ASCII characters will be encoded as a three–character sequence of the form %xx, where xx is the character's hexadecimal value. If the box is unchecked, Fetch will leave names alone (except to ensure that they are not too long for the Macintosh, or contain illegal characters, such as a colon). Unchecked by default.

### **Preferred encoding**

Choose the character set you would like Fetch to use for displaying item names in file lists, saving filenames, and displaying text files. If you often deal with servers that have files in a non–English (or non–Roman) language, this setting may improve the display of filenames. See the [Character Encoding & Translation](#) help topic for more information. **Western (ISO Latin 1)** is the default setting.

These preferences cover unusual situations that most users will not encounter. If you don't understand what the preferences below refer to, don't worry; the default settings are correct for almost all users.



### Use MD5 for S/Key

While the [S/Key](#) standard says that challenge responses should be hashed with MD4, some locations use MD5 for hashes instead. If your site or server is one of those, check this preference so that your responses to challenges will work properly. Unchecked by default.

### Do not send ADAT probe command

Normally Fetch sends the ADAT command to find out if the server supports Kerberos security, so that it can warn you if you aren't taking advantage of security options supported by the server. Servers are supposed to return an error code if they get a command they don't support, so sending this command should be harmless. But some servers, particularly proxy servers, are touchier than they should be and erroneously drop the connection when they get an unfamiliar command. If your server is one of these, you can check this preference to disable sending the ADAT command. Unchecked by default.

### Do not send MACB probe command

The MACB command determines if a server can interpret and generate MacBinary format. Most older Macintosh-based FTP servers can, and taking advantage of that capability makes transfers more reliable on those servers. However, this capability has become less common and some servers do not support the MACB command reliably. If you regularly work with a Macintosh-based FTP server that supports MacBinary format transfers, you may want to uncheck this preference so that Fetch and the server will use MacBinary for all transfers. Otherwise, you should leave this preference checked to avoid problems with servers that give errors when they

receive an unrecognized command, or that support MACB poorly. Checked by default.

### **Do not use MDTM commands**

When mirroring two folders or preserving modification dates for downloads, Fetch sends the MDTM command to the server to get the correct UTC modification date and time of a file. Unfortunately, some servers respond to this command with incorrect information, resulting in incorrect mirroring results or incorrect modification dates. When this preference is checked, Fetch will use the date and time information in the file list, instead of using the MDTM command. Unchecked by default.

### **Send dummy password to Kerberos servers**

When this box is checked, this preference works around a bug in some servers available when this was written. Checked by default.

### **Use "TYPE I" to transfer binary files**

When this box is checked, Fetch will use the "TYPE I" format for binary transfers, which is the commonly supported method, but can cause problems with servers that do not use 8-bit bytes (such as TOPS-20 systems). When this box is unchecked, Fetch will use "TYPE L 8" for binary transfers instead. Checked by default.

### **Do not preserve modification dates**

Normally, when Fetch downloads files from a server, it sets the modification date and time of the downloaded files to be the same as it was on the server. Check this preference to set the modification date and time to be the date and time the file was downloaded instead. Unchecked by default.

### **Recognize and decode AppleSingle**

When this box is checked, Fetch will automatically recognize and convert [AppleSingle](#) files when downloading. If you have trouble downloading a file that you believe is in AppleSingle format, try unchecking this option, downloading the file, and then converting the file manually with Stuffit Expander or another utility. Checked by default.

### **Do not upload dot items in folders**

When this box is checked, Fetch will not upload any files or folders whose names begin with periods (often called "dot items") whenever you upload an entire folder that contains them. Fetch never uploads hidden (invisible) Macintosh files when uploading folders (so for example, .DS\_Store files will never be uploaded, regardless of whether this preference is checked or not). Unchecked by default.

### **Use direct server-to-server transfers**

When this box is checked, Fetch will use direct server-to-server transfers (also known as FXP), which while faster, may not work as reliably as using Fetch as an intermediary for server-to-server transfers. See the [server-to-server transfers](#) help topic for more information. Unchecked by default.

### **Delay between connection retries**

Enter the number of seconds Fetch should wait between attempts to establish a

connection with a server. This preference only applies if you've entered a number in the **Try to connect ... times** field in the [New Connection dialog](#). The default setting is 5.

### **Use text mode transfers for editing**

When this box is checked, when you use the [Edit](#) command, Fetch will download and upload text files using the Text [download mode](#) and [upload format](#). When this box is unchecked, Fetch will use the Binary download mode and Binary (Raw Data) upload format. Generally you want to use the Text mode and format for text files, but if you need to edit files where you don't want line-ending translation to take place (for instance, you have a file with Windows line endings on a UNIX server), you should uncheck this preference. Checked by default.

 [Fetch Help](#) > [Reference](#) > **Windows and Dialogs**

Select one of the topics below for more information about Fetch's windows and dialogs. If you don't see what you're looking for, use the Search field above.

[Edit Shortcut dialog introduction / in-depth](#)

[Fetch Shortcuts](#)

[Fetch Transcript](#)

[Fetch Update \(Single Update\) dialog](#)

[Fetch Update \(Two Updates\) dialog](#)

[File list](#)

[Info window introduction / in-depth](#)

[Media windows](#)

[Message windows](#)

[Mirror window introduction / in-depth](#)

[New Connection dialog introduction / in-depth](#)

[New Shortcut dialog introduction / in-depth](#)

[Preferences](#)

[Resume Download window](#)

[Shortcut list windows introduction / in-depth](#)

[Status pane](#)

[Text windows](#)

[Transfer window](#)

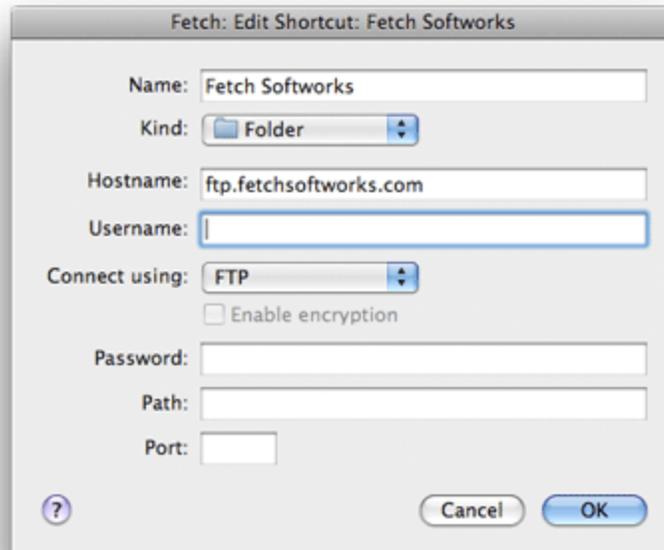
[Verify Certificate dialog](#)

[WebView dialog \(web address only\)](#)

[WebView dialog \(with Fetch address\)](#)



The New Shortcut/Edit Shortcut dialog lets you create a shortcut, or change the [shortcut's](#) information.



Choosing the [New Shortcut](#) or [Edit Shortcut](#) command shows this dialog. The dialog will be filled out for you if you selected a file in a file list and chose **New Shortcut**, or if you chose **Edit Shortcut** to modify an existing shortcut.

Newly created shortcuts are placed in the [Fetch Shortcuts](#) window, unless a different [shortcut list window](#) is active.

The important fields in the dialog are:

**Name** – A name for the shortcut, so you can identify it later.

**Kind** – Whether this shortcut refers to a file or a folder.

**Hostname** – The Internet address of the server the shortcut will access, for example "ftp.fetchsoftworks.com".

**Username** – The username of the account the shortcut should use.

**Connect using** – The type of connection you want to use to access the server (usually **FTP** or **SFTP**). Not all servers support all these choices.

**Password** – The password for the account needed to access the shortcut, if necessary. If you enter a password, it will be saved in your [keychain](#) for future use. If you have previously saved the password for this account in your keychain, you don't need to fill out this field.

**Path** – The location of the file or folder on the server, if desired. If left blank, the shortcut will take you to your home folder on the server.

The name and hostname fields must be filled out. The rest may be left blank if appropriate; for instance, you do not need a username or password for anonymous FTP servers.

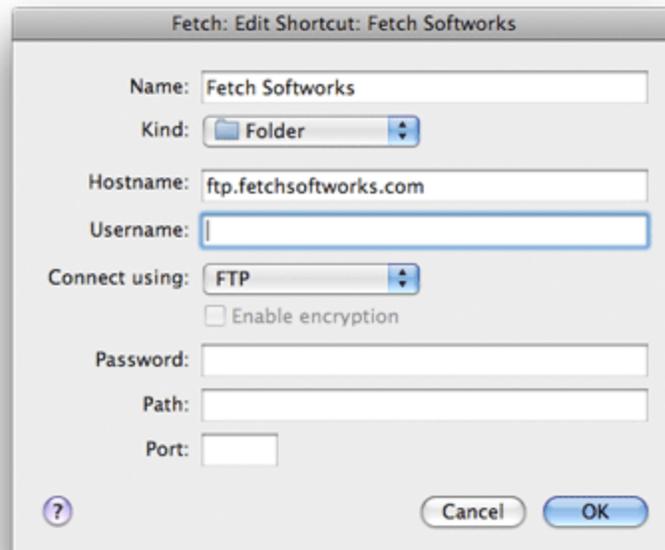
The dialog supports smart pasting of Fetch addresses (FTP [URLs](#)). If you paste a Fetch address, it will be broken into parts and the parts placed in the appropriate fields.

For more information about using the New Shortcut/Edit Shortcut dialog and its contents, see the [New Shortcut/Edit Shortcut dialog in-depth](#) help topic.

## [Fetch Help](#) > [Windows](#) > **New Shortcut/Edit Shortcut Dialog**

The New Shortcut/Edit Shortcut dialog lets you create a shortcut, or change the [shortcut's](#) information.

This is a detailed description of New Shortcut/Edit Shortcut dialog; for a quick guide, see the [New Shortcut/Edit Shortcut introduction](#) help topic.



Choosing the [New Shortcut](#) or [Edit Shortcut](#) command shows this dialog. Depending on how you bring up the dialog, it may already be filled out for you; for instance, if you select a file in a file list and choose **New Shortcut**, information to create a shortcut to that file will be automatically put in the dialog. Just make any changes and give the shortcut a name, and you're done. The information will also be filled out when you choose **Edit Shortcut**.

Newly created shortcuts are placed in the [Fetch Shortcuts](#) window, unless a different [shortcut list window](#) is active.

The information requested in the dialog is:

**Name** – A name for the shortcut, so you can identify it later.

**Kind** – Whether this shortcut refers to a file or a folder. Fetch makes its best guess, but if you know that it has chosen incorrectly, you can pick a different kind. A file shortcut refers to a single remote file. A folder shortcut refers either to a remote folder or a server. If Fetch cannot determine if the shortcut refers to a file or a folder, it will be created as a shortcut of kind "Unknown".

**Hostname** – The Internet address of the server the shortcut will access, for example "ftp.fetchsoftworks.com". You can also enter an IP address (for example "216.92.39.106").

**Username** – The username of the account the shortcut should use. If the shortcut refers to an anonymous FTP server, you do not need to enter a username.

**Connect using** – The type of connection you want to use to access the server. The

choices are **FTP** (no security), **SFTP**, **FTP with TLS/SSL** (FTPS), **FTP with GSSAPI** (Kerberos), and **FTP with KClient** (Kerberos). Not all servers support all these choices; be sure the server you're connecting to supports your choice. See the [connection type](#) and [security](#) help topics for more information about these options.

**Enable Encryption** – Whether to connect to the server using encryption to protect your data transfers or not. This checkbox only applies to **FTP with TLS/SSL** or Kerberos (**FTP with GSSAPI** and **FTP with KClient**) connections. You cannot encrypt data when using plain **FTP** connections, and encryption is always enabled for **SFTP** connections. See the [Security](#) help topic for more information.

**Password** – The password for the account needed to access the shortcut, if necessary. If the shortcut refers to an anonymous FTP server, you do not need to enter a password. If you enter a password, it will be saved in your [keychain](#) for future use. If you have previously saved the password for this account in your keychain, you do not need to fill out this field. If the account requires a password, and you do not enter one when you create the shortcut, you will be prompted for the password when you use the shortcut to connect. This field does not apply to Kerberos connections, for which your Kerberos tickets are used instead.

**Path** – The location of the file or folder on the server, if desired, for example "example/fetchlogo.jpg" for a file, or "mysubfolder" for a remote folder. If left blank, the shortcut will take you to your home folder on the server.

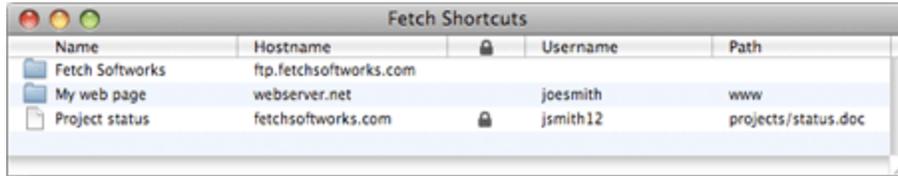
**Port** – If the server the shortcut will connect to uses a non-standard TCP port, enter that port number here. Enter 990 to use SSL connect (Implicit SSL) instead of AUTH TLS when connecting with **FTP with TLS/SSL**. Typically you leave this field blank.

The name and hostname fields must be filled out. The rest may be left blank if appropriate.

The dialog supports smart pasting of Fetch addresses (FTP [URLs](#)). If you paste a Fetch address, it will be broken into parts and the parts placed in the appropriate fields. If you've set up [WebView](#) for a server, you can also paste web addresses (HTTP URLs) for that server and Fetch will fill out the dialog appropriately.

 [Fetch Help](#) > [Windows](#) > **Fetch Shortcuts**

The Fetch Shortcuts window lists the [shortcuts](#) you've created to frequently accessed files, folders, and servers.



To open the Fetch Shortcuts window, choose **Shortcuts > Show Shortcuts** .

Double-click on a shortcut to open the remote folder or download the file it refers to.

The shortcuts in the Fetch Shortcuts window also appear in the Shortcuts  pop-up menu in the [New Connection dialog](#), at the bottom of the **Shortcuts** menu, and in the **Shortcuts** submenu of the Fetch Dock icon's menu. You can close the Fetch Shortcuts window and just use the **Shortcuts** menus instead.

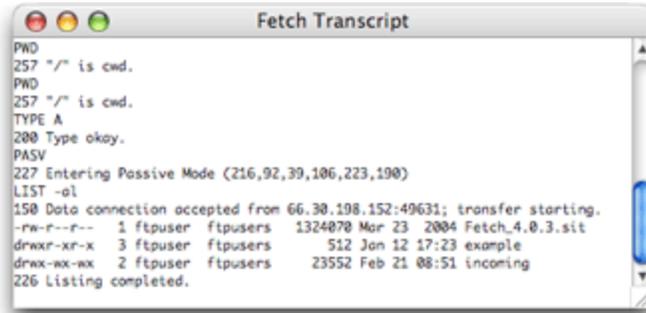
The easiest ways to add a new shortcut to Fetch Shortcuts are to drag a file or folder from a transfer window to it, or to choose **Make Shortcut** from the Shortcuts pop-up menu in the New Connection dialog after filling in the fields of that dialog. For more information and detailed steps, see the [creating a shortcut](#) help topic.

To delete a shortcut, select it and press the Delete key, or drag it to the Mac OS Trash. For more information, see the [deleting or changing a shortcut](#) help topic.

The Fetch Shortcuts window is a special [shortcut list window](#) that is automatically saved and automatically reopened when you start Fetch. For more information about working with the Fetch Shortcuts window, see the [shortcut list windows](#) help topic. Also see the [shortcuts](#) topic for more general information about shortcuts.

## [Fetch Help](#) > [Windows](#) > Fetch Transcript

The Fetch Transcript window shows a complete, unfiltered record of Fetch's interactions with FTP or SFTP servers — all of the commands Fetch sends, and the servers' responses.



```
PWD
257 "/" is cwd.
PWD
257 "/" is cwd.
TYPE A
200 Type okay.
PASV
227 Entering Passive Mode (216,92,39,106,223,190)
LIST -ol
150 Data connection accepted from 66.30.198.152:49631; transfer starting.
-rw-r--r--  1 ftpuser  ftpusers  1324878 Mar 23  2004 Fetch_4.0.3.sit
drwxr-xr-x  3 ftpuser  ftpusers    512 Jan 12 17:23 example
drwx-wx-wx  2 ftpuser  ftpusers   23552 Feb 21 08:51 incoming
226 Listing completed.
```

The transcript contains a lot of technical and extraneous information that you usually do not need to see; Fetch interprets the responses from the server to show you the transfer window and file list in an easy-to-use format. However, when you are having trouble with a connection or transferring files, the Fetch Transcript window can provide useful debugging information.

Some servers display special information when the connection is established (such as hours of operation, use restrictions, and number of open connections). This information is also displayed in the Fetch Transcript window. You can display this information in separate windows by setting the **Display server messages** preference in the [General preferences](#) pane to **in transcript and in separate windows**.

Extended file lists with complete details appear in the Fetch Transcript window, however, you can more easily view those by using the **Remote** > [View Verbose File List](#) command.

You can save the contents of the Fetch Transcript to a file with the **File** > **Save a Copy As** command, or select text in the window and copy it to the clipboard to be pasted elsewhere, just as you can with any other [text window](#) in Fetch.

There is only one Transcript window. If you have multiple connections open, the commands to and responses from various servers may be mixed together. If you are trying to troubleshoot a problem, close all but the problem connection to simplify reading of the Transcript window.

Sometimes when providing support for Fetch, we may ask you to send a transcript to Fetch Softworks to help us debug the problem. See the [sending a transcript to Fetch Softworks](#) help topic for step-by-step instructions.



## [Fetch Help](#) > [Windows](#) > Fetch Update (Single Update) Dialog

When Fetch detects that a new version of Fetch is available, either from the automatic update check or when you choose [Check for Update](#) manually, it displays this dialog. The Fetch Update dialog tells you the version number of the update, whether the update is free or requires a purchase, and presents you with several possible actions.

You can click the **More Info** button to see a webpage with more information about this update, such as a description of new features and other changes. After reading the details about the update, you can click **Quit & Get Update** or **Buy** (depending on whether the update is free or requires a purchase) to download the update. If you don't want to download and install the update now, you can click **Later** to be prompted again later, or **Skip** to not be reminded about the update ever again.

More details:

**Quit & Get Update** – (Only present for free updates.) Click this button to download the update in your web browser. Fetch will quit before the download begins, because if your current copy of Fetch is running, you cannot replace it with the update. If you have downloads in progress or have windows with unsaved changes, Fetch will ask you to review the windows before you quit. Fetch will then tell your web browser to download the update. Once the download is finished, you should mount the disk image (if it isn't mounted automatically) and drag the new Fetch to the Applications folder or your preferred location for it. If the download is not successful, relaunch Fetch, choose **Fetch > Check for Update**, and try again.

**Buy** – (Only present for updates requiring a purchase.) Click this button to be guided through purchasing the update. Once your purchase is complete, you will be prompted to download the update in your web browser.

**More Info** – Click this button to see a webpage with more information about the Fetch update being offered to you, such as a description of new features and other changes.

**Later** – Click this button to dismiss the Fetch Update dialog without any action being taken. You will be reminded about this update again the next time an automatic update check takes place. You can also get the update at a more convenient time by choosing **Fetch > Check for Update** manually.

**Skip** – Click this button to tell Fetch that you do not want to be reminded about this update again. You will not be reminded about this update when the next automatic update check takes place; the next time you'll be reminded will be when a further update to Fetch is available. However, if you later change your mind and want to get this update, you can choose **Fetch > Check for Update**, and you will be offered the currently available updates. You can also go to the [Fetch Softworks](#) website to download the update.

In some cases, there may be two updates available for Fetch at the same time – a free update and an update that requires a purchase. When two updates are available, Fetch will show the [Fetch Update \(Two Updates\)](#) dialog instead.

Related topics:

- [Software Update](#)
- **Automatically check for updates** preference in the [Miscellaneous preferences](#)
- [Check for Update](#)
- [Fetch Update \(Two Updates\) dialog](#)



## [Fetch Help](#) > [Windows](#) > **Fetch Update (Two Updates) Dialog**

When Fetch detects that two updates – both a free update and an update that requires a purchase – are available, either from the automatic update check or when you choose [Check for Update](#) manually, it displays this dialog. The Fetch Update dialog tells you the version numbers of both updates, allows you to get information about either update, and install one or buy the other.

When only one update is available, Fetch will show the [Fetch Update \(Single Update\)](#) dialog instead.

Next to the name of each update is a **More Info** button. Click either button to see a webpage with more information about that Fetch update, such as a description of new features and other changes. After reading the details about each update, click the radio button next to the update you'd like to download. You will be given a choice to **Quit & Get Update** or **Buy**, depending on your selection. If you don't want to install either update now, you can choose **Later** to be prompted again later, or **Skip Both** to not be reminded about either update.

More details:

**Quit & Get Update** (for free updates) – Click this button to download the free update in your web browser. Fetch will quit before the download begins, because if your current copy of Fetch is running, you cannot replace it with the update. If you have downloads in progress or have windows with unsaved changes, Fetch will ask you to review the windows before you quit. Fetch will then tell your web browser to download the update. Once the download is finished, you should mount the disk image (if it isn't mounted automatically) and drag the new Fetch to the Applications folder or your preferred location for it. If the download is not successful, relaunch Fetch, choose **Fetch > Check for Update**, and try again.

**Buy** (for updates requiring a purchase) – Click this button to be guided through purchasing the update. Once your purchase is complete, you will be prompted to download the update in your web browser.

**Later** – Click this button to dismiss the Fetch Update dialog without any action being taken. You will be reminded about both updates again the next time an automatic update check takes place. You can also get either update at a more convenient time by choosing **Fetch > Check for Update** manually.

**Skip Both** – Click this button to tell Fetch that you do not want to be reminded about either update again. You will not be reminded about either update when the next automatic update check takes place; the next time you'll be reminded will be when a further update (either free or requiring a purchase) to Fetch is available. However, if you later change your mind and want to get either update, you can choose **Fetch > Check for Update**, and you will be offered the currently available updates. You can also go to the [Fetch Softworks](#) website to download either update.

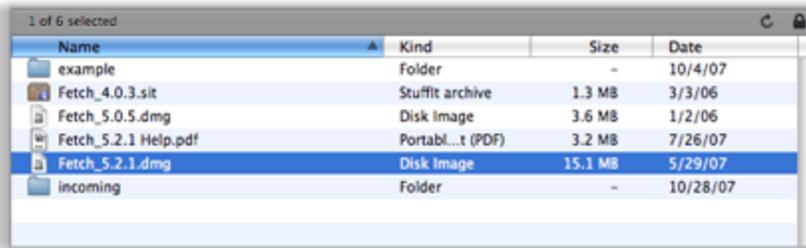
If you choose to download the free update, the next time an update check is done, Fetch will remind you again about the update that requires a purchase. You can either click **Later** or **Skip** when that happens.

Related topics:

- [Software Update](#)
- **Automatically check for updates** preference in the [Miscellaneous preferences](#)

- [Check for Update](#)
- [Fetch Update \(Single Update\) dialog](#)

The list of file and folders names in the middle of a Fetch [transfer window](#) is the file list. It works much like the Mac OS Finder's list view.



In the file list, clicking an item selects it. Many commands such as **Get**, **Delete Item**, **Edit**, etc. operate on the selected item. Holding down the Command key while clicking allows you to select multiple individual items, and holding down the Shift key while clicking selects a range of items. You can also select a filename by typing the first few letters of its name, or using the up and down arrow keys to step through the list.

Double-clicking an item prompts Fetch to either download the file or change to the selected folder. Pressing Command-Down arrow is equivalent to double-clicking. Command-double-click a folder to display its contents in a new transfer window.

Dragging files or folders from the Mac OS Finder to the file list will upload those items to the current folder displayed in the file list; if you drag them to a folder icon in the Fetch file list, they will be uploaded to that remote folder.

Dragging files or folders from the file list to the Mac OS Finder will download the items. Dragging items in the file list to a folder in the file list will move those items on the server. Dragging items from one file list to another file list will either move the items (if the file lists are for folders that are on the same server) or copy the items to the other server (if the file lists are for folders on different servers.)

You can rename files in the file list by clicking once to select an item, and then pressing the Return key. An edit box appears around the name. Type a new name or edit the existing name, and press Return again when you're done. For more information, see the [renaming files and folders](#) help topic.

The window title bar shows the name of the current folder, and then either the name of the shortcut matching this server or the server's hostname. Click the **Path** button in the toolbar or Command-click the window's title to display the complete hierarchy leading to the current folder. Choose a folder from the menu to make that the current folder and update the file list.

The **Back** button in the toolbar takes you back to the folder you were previously viewing. See the [changing folders](#) help topic for other ways to change the current folder and move between folders.

The file list can be sorted by name, file kind, file size, or last modification date. The current sort order is indicated by the highlighted column title. To change the sort order, click a different column title. Click a highlighted column title to reverse the sort order. The columns can be resized by clicking on the divider between the titles and dragging. You can also reorder the columns by clicking in the middle of a column heading and dragging it to the left or right.

Fetch displays a small icon next to each name. Files have an icon appropriate for the kind of

file they are; if Fetch cannot determine the kind of a file, a standard document icon is shown. Folders have the standard folder icon. If Fetch cannot determine whether an item is a file or folder, a combined folder/document icon is displayed. If the item is a UNIX symbolic link (a pointer to a file or folder in a different place), Fetch shows an alias badge on the icon.

The info bar just above the file list contains the following items, from left to right:

- The total number of items in the file list.
- A date and time showing when the file list was last read from the server, if the file list was read from the cache, as explained below.
- The **Refresh**  button, which reloads a fresh file list from the server. If you don't see a file listed in the file list after uploading it, click the **Refresh** button.
- For secure connections ([SFTP](#), [FTP with TLS/SSL](#), and [Kerberos](#) with encryption), Fetch displays a locked padlock icon. An unlocked padlock icon indicates your password was sent securely, but encryption is not being used to secure your transfers. No padlock icon is displayed if the connection is not secure at all. For more information, see the [padlock icon](#) and [security](#) help topics.

Fetch stores file lists in the "Fetch Cache" file in your Preferences folder. For increased speed, Fetch reuses file lists rather than downloading them again when there is reason to believe that the stored list is up-to-date. When Fetch uses a stored file list, it shows the date and time when the list was originally retrieved on the right just above the file list. If the stored list seems to be out of date, you can reload a fresh list from the server by choosing **View > Refresh** or clicking the **Refresh**  button. (The technical details: a file list is cached for 5% of the age of its newest item. For example, if newest item's modification time is 20 days ago, the list will be read from the cache for the following day.)

Useful keyboard shortcuts for the file list: Pressing the Down arrow key while holding down the Command key makes the selected folder the current folder. Pressing Command-Up arrow takes you to the parent folder (go up one level). You can also use the Home, End, Page Up, and Page Down keys to scroll the file list.

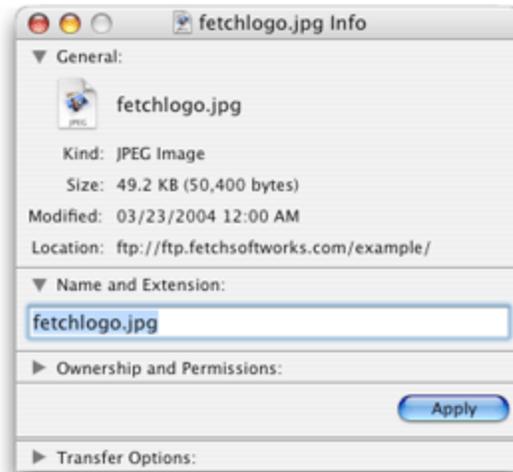
You can also perform common transfer window actions with the [file list contextual menu](#). Control-click an item in the file list to display the contextual menu. The file list contextual menu includes two special commands not available elsewhere — the **Duplicate** command for duplicating files on the server, and the **Clone Window** command. Both open another connection and transfer window to the same server of the current transfer window.

It's also possible to make a new copy of an item in one folder by Option-dragging the item to another folder icon in the file list. The copy  cursor indicates that the item will be copied rather than simply moved. Like the **Duplicate** command, Option-dragging an item to make a copy of it will open a second connection to the server in a new transfer window.

Another way to copy files and folders from one folder to another on the same server, or from one server to another, is to use the **Copy** and **Paste** commands. See the [copying files using copy and paste](#) help topic for more information.

## [Fetch Help](#) > [Windows](#) > **Info Window Introduction**

Info windows display detailed information about remote files and folders. You can also rename items, and set permissions and transfer options for files.



To display an info window, select an item (or items) in a file list and choose **Remote > Get Info**.

Info windows are divided into several panes: General, Name and Extension, Ownership and Permissions, and Transfer Options. You can hide and show each pane by clicking the disclosure triangle to the left of its name.

The General pane shows the selected item's icon, kind, size, modification date, and location. If you selected multiple items, it shows the number of files and folders selected. If you selected a folder, or multiple items that include a folder, there is a button to calculate the size of all of the selected items (since calculating the size of folders can be time consuming, Fetch does not do it automatically).

To change a file or folder's name or extension, edit the field in the Name and Extension pane, then click the **Apply** button.

The Ownership and Permissions pane shows who has access to read (view) and write (modify) the selected items. To change the items' permissions, check or uncheck the checkboxes, then click the **Apply** button. You can also edit the permissions using the UNIX equivalent edit field. For more information about interpreting and changing permissions, see the [permissions](#) and [viewing and setting permissions](#) help topics.

To discard any changes to the name or permissions, just close the info window without clicking the **Apply** button.

The Transfer Options pane lets you tell Fetch how it should download a file, and what it should do with it afterwards — such as whether to automatically open or decode files, or which application will be used to open or edit the file. For more information, see the [transfer options](#) help topic.

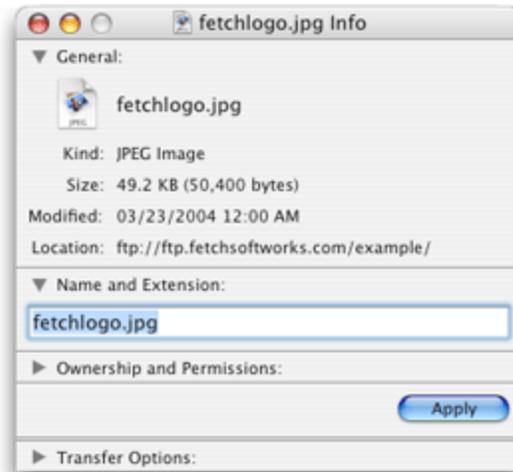
The panes you see in the info window will vary depending on whether you selected a single file, a single folder, or multiple files or folders.

For more information about using the info window, see the [info window in-depth](#) help topic.



## [Fetch Help](#) > [Windows](#) > **Info Window**

Info windows display detailed information about remote files and folders. You can also rename items, and set permissions and transfer options for files.



This is a detailed description of the info window; for a quick guide, see the [info window introduction](#) help topic.

To display an info window, select an item (or items) and choose **Remote > Get Info**.

Info windows are divided into several panes: General, Name and Extension, Ownership and Permissions, and Transfer Options. You can hide and show each pane by clicking the disclosure triangle to the left of its name. Which panes you see in the info window will vary, depending on whether you selected a single file, a single folder, or multiple files or folders.

The General pane shows the selected item's icon, kind, size, modification date, and location. If you selected multiple items, it shows the number of files and folders selected. If you selected a folder, or multiple items that include a folder, there is a button to calculate the size of all of the selected items (since calculating the size of folders can be time consuming, Fetch does not do it automatically).

Certain special characters in the location field may be percent-encoded (for example, %2F for a slash or %40 for an at sign) to conform with the FTP URL specification.

To change a file or folder's name or extension, edit the field in the Name and Extension pane, then click the **Apply** button.

The Ownership and Permissions pane shows who has access to read (view) and write (modify) the selected items. Permissions are shown both as checkboxes and a UNIX equivalent.

To change permissions, check or uncheck the checkboxes, then click the **Apply** button. If you have selected multiple items, some of the checkboxes may have a dash instead of being checked or unchecked. That means not all of the items have the same permissions. Leaving a checkbox with a dash unchanged will preserve the mixed permissions, even if you change other permissions. Changing a folder's permissions does not set the permissions of its contents — it only changes the permissions of the folder itself.

For just uploaded items on FTP connections, Fetch may show all the permissions checkboxes as unchecked and the owner and group as "unknown". This is not the true state of the

permissions. To see the true permissions, you must click the [Refresh](#)  button in the transfer window, and get info on the items again. You can set the permissions immediately without refreshing. On SFTP connections, you will see the true permissions immediately, but you must refresh to see the true owner and group.

Typically you can only see or view permissions on servers running the UNIX operating system. For more information about interpreting and changing permissions, see the [permissions](#) and [viewing and setting permissions](#) help topics.

To discard any changes to the name or permissions, just close the info window without clicking the **Apply** button.

The Transfer Options pane lets you tell Fetch how it should download a file, and what it should do with it afterwards — such as whether to automatically open or decode files, or which application will be used to open or edit the file. For more information, see the [transfer options](#) help topic.

Command-click the proxy icon or title in an info window title bar to show the path to the items described in the window. Choosing a folder from the path pop-up menu shows that folder in a transfer window. Drag the proxy icon to a shortcut list window to create a shortcut to the items, to the Finder to create a [droplet shortcut](#) (for folders) or a [shortcut document](#) (for files), or to a text editor to copy the items' Fetch address (URL).

For a single file, the info window displays the file's icon, kind, exact size, modification date, location, name and extension, ownership, permissions, and transfer options.

For a single folder, the info window displays the folder's modification date, location, name, and permissions. There is a **Calculate** button that calculates the exact size of the folder's contents and how many files and folders it contains.

For multiple items, the info window shows the total number of items selected, the location of the items, and permissions. You can set the permissions of all the selected items, which is an easy way to make sure a group of files has the same permissions. If only multiple files are selected, the total size of all files is displayed automatically. If the items include a folder, there is a **Calculate** button that calculates the total size of all items (including the size of all folders' contents). For multiple folders or folders containing many items, calculating the size may take some time.

The permissions checkboxes may all be unchecked, and the ownership listed as "unknown", for files that you have just uploaded using FTP. Click the [Refresh](#)  button in the transfer window and get info on the items again to see the true permissions.

The Transfer Options pane is not shown when you get info for an alias.

You can use the info window to find out how much space your files are taking up on your server, or how close you are to your server storage limit. When you select a set of files and folders, choose **Get Info**, and click the **Calculate** button, Fetch calculates the disk space taken up not only by those items, but by the items inside the folders too. Either select all the files and folders in a remote folder, or select the parent folder of the items in question, choose **Get Info** and click **Calculate**. The combined size of all items will be displayed in the info window. For many items, or folders with a large number of items inside them, this may take a while.

 [Fetch Help](#) > [Windows](#) > **Media Windows**

Media windows display the contents of image, movie, sound, PDF, and other media files on a server when you use the [View](#) command on Mac OS X 10.4. (Fetch media windows are not used by the **Quick Look** command on Mac OS X 10.5 and later.)

Any files that can be displayed by Apple's QuickTime software can be displayed in a media window. For many types of media files, such as movies, sounds, and PDF files, the media window will show the standard QuickTime controller at the bottom of the window.

In any media window you can:

- Print the contents of the window with the [Print](#) command.
- Save the contents of the window to a file on a local drive with the [Save a Copy As](#) command.

You can edit a remote image file on a server using the [Edit](#) command. If you want to make changes to a media file locally but not on the server, use the [Save a Copy As](#) command to save it to disk and then edit it with your favorite editor for that type of media.

The **Quick Look** command, which takes the place of **View** on Mac OS X 10.5 and later, does not use Fetch media windows; for information about the windows displayed by Quick Look, see the [Quick Look](#) help topic.

 [Fetch Help](#) > [Windows](#) > **Message Windows**

Message windows are [text windows](#) that contain special information displayed by some servers when you open a connection. Message windows usually contain information such as hours of operation, use restrictions and rules, number of open connections, and information about the files on the server.

Separate message windows are only displayed when the **Display server messages** preference in the [General Preferences](#) pane is set to **in transcript and in separate window**.

Messages from the server are always displayed in the [Fetch Transcript](#) window, regardless of whether you've chosen to display them in separate windows or not.

You can save the contents of message windows to a file with the **File > Save a Copy As** command, or select text in the window and copy it to the clipboard to be pasted elsewhere, just as you can with any other [text window](#) in Fetch.

## [Fetch Help](#) > [Windows](#) > Mirror Window Introduction

The Mirror window lets you automatically copy all new or changed files from a local Macintosh folder to a remote folder on a server, or vice versa.



For example, if you keep all the files that make up a website in a folder on your Macintosh, you can use the Mirror window to quickly upload only the files that are new or have changed since your last update, instead of using the [Put](#) command to upload the entire site again. You can also save a [mirror document](#) that makes uploading the website changes a simple one-click process.

You can display the Mirror window by choosing **Remote > Mirror**, or you can add a **Mirror** button to the [transfer window toolbar](#).

The Mirror window has two panes — one for specifying the source and one for specifying the destination. Mirroring can take place between a folder on your computer and a folder on a server, or two server folders. In the left-hand pane, you can choose either a local or remote folder by clicking the appropriate tab; the right-hand pane always refers to a remote folder.

You can specify a local folder by clicking the **Choose** button, which shows the standard choose folder prompt, or by dragging a folder icon from the Finder to the left-hand pane of the Mirror window.

You can specify remote folders by choosing a currently open connection from the pop-up menu to the left of the Host field, or by clicking the **Choose** button. The **Choose** button displays a dialog similar to the [New Connection dialog](#). Fill out the dialog with the information for the desired folder on the server.

Click one of the two arrow buttons between the two panes to indicate the direction in which files should be copied.

The [Delete stray items at destination](#) checkbox instructs Fetch to delete items in the destination folder that do not exist in the source folder, letting you synchronize the two folders. **This is a dangerous option, since it can delete entire folders and their contents.** Be very sure when you select it, and use it with care!

Click the **Mirror** button to start mirroring.

Note that the first time you mirror a local folder to a remote folder, Fetch will most likely upload all your files, because it needs to start by synchronizing all the files. But the next and all subsequent times you use the Mirror window for the same folders, only new or changed

files will be uploaded.

The **Save** button saves the current Mirror window settings to a [mirror document](#). Opening that document will open the Mirror window with those settings, or optionally immediately start mirroring.

For more information, see the [Mirror window in-depth](#) topic.

## [Fetch Help](#) > [Windows](#) > Mirror Window

The Mirror window lets you automatically copy all new or changed files from a local Macintosh folder to a remote folder on a server, or vice versa. You can also use it to synchronize a Macintosh folder with a server folder.

This is a detailed description of the Mirror window; for a quick guide, see the [Mirror window introduction](#) help topic.



For example, if you keep all the files that make up a website in a folder on your Macintosh, you can use the Mirror window to quickly upload only the files that are new or have changed since your last update, instead of using the [Put](#) command to upload the entire site again. You can also save a [mirror document](#) that makes uploading the website changes a simple one-click process.

You can display the Mirror window by choosing **Remote > Mirror**, or you can add a **Mirror** button to the [transfer window toolbar](#).

The Mirror window has two panes — one for specifying the source and one for specifying the destination. Mirroring can take place between a folder on your computer and a folder on a server, or two server folders. In the left-hand pane, you can choose either a local or remote folder by clicking the appropriate tab; the right-hand pane always refers to a remote folder.

You can specify a local folder by clicking the **Choose** button, which shows the standard choose folder prompt, or by dragging a folder icon from the Finder to the left-hand pane of the Mirror window. Once you've chosen a local folder, the path to it will be displayed in the pane.

You can specify remote folders in several ways:

- The pop-up menu to the left of the Host field lists the folders represented by currently open [transfer windows](#). Choose one from the menu to specify it as the remote folder to use.
- Drag a remote folder from a transfer window into one of the Mirror window panes.
- The **Choose** button displays a dialog similar to the [New Connection dialog](#). Fill out the dialog with the information for the desired folder on the server.

Once you've chosen a remote folder, the hostname, username, and path to the remote folder will be displayed in the pane. Fetch will display a [padlock icon](#) if you've chosen to connect securely (using [SFTP](#), [FTP with TLS/SSL](#) or [Kerberos](#) with encryption) to the server.

Click one of the two arrow buttons between the two panes to indicate the direction in which files should be copied.

The [Delete stray items at destination](#) checkbox instructs Fetch to delete items in the destination folder that do not exist in the source folder, letting you synchronize the two folders. **This is a dangerous option, since it can delete entire folders and their contents.** Be very sure when you select it, and use it with care!

Click the **Mirror** button to start mirroring.

Note that the first time you mirror a local folder to a remote folder, Fetch will most likely upload all your files, because it needs to start by synchronizing all the files. But the next and all subsequent times you use the Mirror window for the same folders, only new or changed files will be uploaded.

The **Save** button saves the current Mirror window settings to a [mirror document](#). Opening that document will open the Mirror window with those settings, or optionally immediately start mirroring.

When you click the **Save** button, you will be prompted for the name and location for the mirror document. Check the **Mirror automatically when opened** box to tell Fetch to automatically start mirroring when you open the mirror document. Otherwise, Fetch will open the Mirror window filled out with the folders specified in the document, and you will have to click the **Mirror** button to start mirroring.

If the connections saved in a mirror document require passwords, you will be prompted for them when mirroring starts. If you do not want to be prompted for passwords, you need to save the passwords for the connections in your [keychain](#).

## [Fetch Help](#) > [Windows](#) > **New Connection Dialog Introduction**

The New Connection dialog is where you tell Fetch which server you want to transfer files from or to.



Fetch displays the New Connection dialog automatically when you start Fetch. You can show it at other times by choosing **File > New Connection**.

The dialog requests the following information from you:

- Hostname: the name of the server you want to transfer files from or to
- Username: the name of your account on that server
- Connect using: how you're going to connect to the server (usually FTP or SFTP)
- Password: the password for your account

The hostname of the server is required. If the username is left blank, the special username "anonymous" will be sent to the server (but not displayed). This indicates you want anonymous (guest) access to an FTP site. When connecting anonymously it is not necessary to enter a password.

If you're not sure what information to enter in the New Connection dialog, see the [What do I enter in the hostname, username, and password fields?](#) help topic.

**Note:** The information entered in the New Connection dialog has nothing to do with your license to use Fetch. It specifies the server with which you'd like to transfer files. The people who operate the server you are trying to connect to (such as your Internet service provider or hosting provider) should provide you with the information you need to connect to their FTP server, SFTP server, or web server.

You can also fill out some or all of these fields by choosing a recent connection from the Recent Connections  pop-up or a shortcut from the Shortcuts  pop-up menu.

To learn how to create your own shortcut, see the [creating a shortcut](#) help topic.

Fetch always remembers your last choice from the **Connect Using** menu for the next time you use the New Connection dialog; this is handy if you always use the same connection type.

If you click the disclosure  button, the dialog expands to reveal additional fields. You can

learn about them in the [New Connection dialog in-depth](#) topic.

You can make the New Connection dialog open with connection information already filled out for you by creating or specifying a default shortcut. See the [making Fetch start with information for your server or website](#) help topic for more information.

The dialog supports smart pasting of [Uniform Resource Locators \(URLs\)](#). If you paste a URL from another application, it will be broken into parts and the parts placed in the appropriate fields.

For more information about using the New Connection dialog and its contents, see the [New Connection dialog in-depth](#) help topic.

## [Fetch Help](#) > [Windows](#) > **New Connection Dialog**

The New Connection dialog is where you tell Fetch which server you want to transfer files from or to.

This is a detailed description of the New Connection dialog; for a quick guide, see the [New Connection dialog introduction](#) help topic.



Normally Fetch displays the New Connection dialog automatically when you start Fetch. You can show it at other times by choosing **File > New Connection**.

You can disable the automatic display of the New Connection dialog by unchecking the **Show New Connection dialog at startup** preference in the [General Preferences](#) pane.

The dialog requests the following information from you:

- Hostname: the name or IP address of the server you want to transfer files from or to
- Username: the name of your account on that server
- Connect using: how you're going to connect to the server (usually FTP or SFTP)
- Password: the password for your account

The hostname of the server is required. If the username is left blank, the special username "anonymous" will be sent to the server (but not displayed). This indicates you want anonymous (guest) access to an FTP site. When connecting anonymously it is not necessary to enter a password.

If you click the disclosure  button, the dialog expands to reveal additional fields. A complete description of each field is below.

If you're not sure what information to enter in the New Connection dialog, see the [What do I enter in the hostname, username, and password fields?](#) help topic.

**Note:** The information entered in the New Connection dialog has nothing to do with your license to use Fetch. It specifies the server with which you'd like to transfer files. The people who operate the server you are trying to connect to (such as your Internet service provider or hosting provider) should provide you with the information you need to connect to their FTP server, SFTP server, or web server.

You can also fill out some or all of these fields by choosing a recent connection from the Recent Connections  pop-up or a shortcut from the Shortcuts  pop-up menu. You can create your own shortcut by filling out the fields in the New Connection dialog, and then choosing **Make Shortcut** from the Shortcuts pop-up menu. You will be prompted for a name to give the shortcut, and whether it should be the default shortcut. After you connect, the shortcut will be created and appear in the Shortcuts pop-up menu and in the [Fetch Shortcuts](#) window. See the [creating a shortcut](#) topic for other ways to create shortcuts in Fetch.

You may also see [Bonjour FTP and SFTP servers](#) listed in the Shortcuts pop-up menu.

You can specify a shortcut as the default shortcut; the New Connection dialog will always open with information from the default shortcut already filled out. You can set a default shortcut using the **Default shortcut** preference in the [General Preferences](#) pane. You can also create a new default shortcut in the New Connection dialog by filling in the desired information, choosing **Make Shortcut** from the Shortcuts pop-up menu, and checking the **Make this the default shortcut** box.

You can cycle through the shortcuts by pressing Command-] or Command-[. You can cycle through the recent connections while in the dialog by pressing Command-Option-] or Command-Option-[. (In the French localization, these keyboard shortcuts are Command-), Command-(, Command-Option-), and Command-Option-(, respectively.)

The dialog supports smart pasting of Fetch addresses (FTP [URLs](#)). If you paste a Fetch address, it will be broken into parts and the parts placed in the appropriate fields. If you've set up [WebView](#) for a server, you can also paste web addresses (HTTP URLs) for that server and Fetch will fill out the dialog appropriately.

The information requested by the dialog is:

**Hostname** – The Internet address of the server to connect to, for example "ftp.fetchsoftworks.com". You can also enter an IP address (for example "216.92.39.106").

**Username** – The username of the account to connect with, if necessary. If you are connecting to an anonymous FTP server, you do not need to enter a username.

**Connect using** – The type of connection you want to use to access the server. The choices are **FTP** (no security), **SFTP**, **FTP with TLS/SSL** (FTPS), **FTP with GSSAPI** (Kerberos), and **FTP with KClient** (Kerberos). Not all servers support all these choices; be sure the server you're connecting to supports your choice, or you will receive an error. See the [connection types](#) and [security](#) help topics for more information about these choices. Fetch remembers your last choice from this menu for the next time you use the New Connection dialog; this is handy if you always use the same connection type.

**Enable Encryption** – Whether to connect to the server using encryption to protect your data transfers or not. This checkbox only applies to **FTP with TLS/SSL** and Kerberos (**FTP with GSSAPI** and **FTP with KClient**) connections. You cannot encrypt data when using plain **FTP** connections, and encryption is always enabled for **SFTP** connections. See the [Security](#) help topic for more information.

**Password** – The password for your account on the server, if necessary. If you are connecting to an anonymous FTP server, you do not need to enter a password. If you have previously saved the password for this account in your [keychain](#), this field will be automatically filled out. This field does not apply to Kerberos

connections, where your Kerberos tickets are used instead.

**Add to keychain** – When this checkbox is checked, Fetch will securely save the password for the specified server and username in your keychain for future use, so that you do not have to enter it again. For more information, see the [Keychain support](#) help topic.

Click the disclosure  button to see these additional fields:

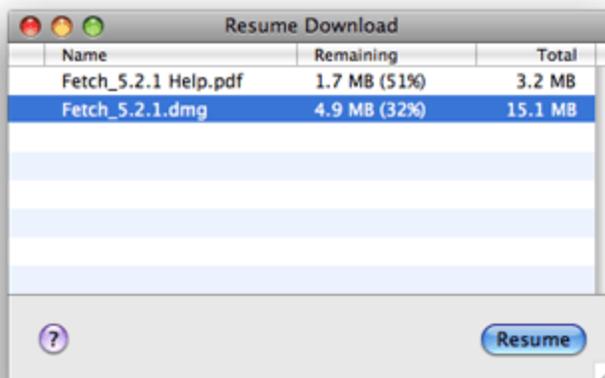
**Initial folder** – If you enter a folder name or a path to a folder on the server, Fetch will automatically go to that folder when you open the connection. If left blank, Fetch will take you to your home folder on the server.

**Port** – If the server uses a non-standard TCP port for connections, enter that port number here. Enter 990 to use SSL connect (Implicit SSL) instead of AUTH TLS when connecting with **FTP with TLS/SSL**. Typically you leave this field blank.

**Try to connect ... times** – If you'd like Fetch to repeatedly try to connect to a busy server, you can specify the number of times to retry in this field.

## [Fetch Help](#) > [Windows](#) > Resume Download Window

The Resume Download feature of Fetch allows you to resume a partially completed download, which can save time compared to starting the whole download over. This is handy if you accidentally stop one before it completes. Since Fetch tries to detect stalls and network or server errors and automatically resume downloads for you, you should not need this feature under most circumstances.



The Resume Download window shows a list of the downloads that can be resumed. To resume a transfer, select its name and then click the **Resume** button, or double-click its name.

An interrupted transfer also leaves a partial file in your download folder or the location you dragged the file to originally. You can also resume a download by double-clicking the partial file in the Finder.

The window lists the name of each partially downloaded file, the amount of the file remaining to download (by actual size and percentage), and the total size of each file. You can sort the list by clicking any column heading, or you can sort the list chronologically by clicking the leftmost unlabeled column heading.

To make Fetch forget about a partial download, select a file from the Resume Download list and press the Delete key, or choose the **Edit > Delete** menu command.

The Resume Download window is displayed when you choose the [Resume Download](#) command.

There are a number of limitations on this feature. Fetch can only successfully resume a download if:

- The file has not changed since you first tried to download it. Otherwise, you will get part of one file and part of another, with unpredictable results.
- The server supports the restart (REST) command.
- If the file is a text file, the server implements the restart command in the way that Fetch expects. Most servers support restarting transfers. When they do, they implement restarted text downloads in one of two different ways, and this can lead to missing or duplicated information in resumed text downloads. You can avoid this problem by downloading in binary mode (choose **Binary** from the [Remote](#) > [Mode](#) submenu), but that may give you a file with incorrect line endings.
- If you were downloading more than one file or a folder when the download stopped, the

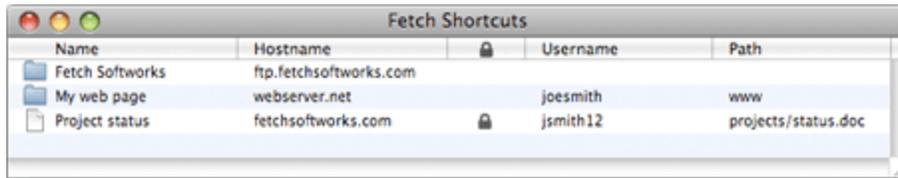
Resume Download feature can only finish downloading the file that was being transferred at the time — Fetch does not remember the list of files it was going to download, but hadn't gotten to yet. You will have to download any other untransferred files again manually. (However, Fetch's automatic resume download feature does work to finish downloads of folders or multiple files.)

Related topics:

- [Resuming an interrupted download](#)
- [Resuming uploads](#)

## [Fetch Help](#) > [Windows](#) > **Shortcut List Windows Introduction**

A shortcut list window collects a group of [shortcuts](#) in a single window. This list of shortcuts can be saved and opened again later.



The [Fetch Shortcuts](#) window, which Fetch saves and opens automatically, is a shortcut list window. Other shortcut list windows are not saved or reopened automatically, and their shortcuts do not appear in the [New Connection dialog](#) or the **Shortcuts** menu. If you expect to use a shortcut often, you should add it to the Fetch Shortcuts window, not a separate shortcut list window.

To create a shortcut list window, choose **Shortcuts > New Shortcut List** .

To open a shortcut in a shortcut list window, double-click it, or select it and press Command-Down arrow. This will either open the remote folder or start downloading the file it refers to. You can also drag a shortcut to a Finder window or the desktop to start a download.

Uploads can be started by dragging files or folders to a shortcut for the destination. Fetch will automatically open a connection to the location referenced by the shortcut and start the upload.

The most common ways to add shortcuts to a shortcut list window are either the **Shortcuts > New Shortcut** command, or by dragging files or folders from a transfer window into a shortcut list window. See the [in-depth](#) help topic for more ways to add shortcuts.

To remove a shortcut from a shortcut list window, select the shortcut and then choose **Edit > Delete**, press the Delete key, or drag the shortcut to the Mac OS Trash icon.

You can sort the shortcuts by clicking any column heading; click the column heading a second time to reverse the sort order.

The entire shortcut list can be saved as a [shortcut list document](#) for future use with **File > Save** or **File > Save As**, and reopened with **File > Open** .

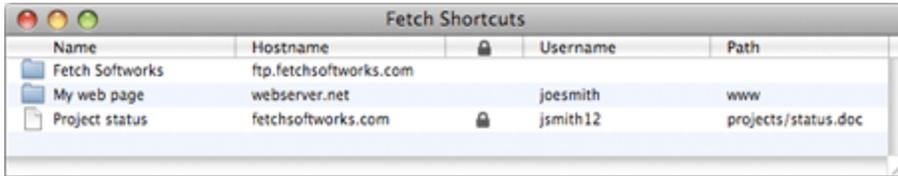
To edit a shortcut in a shortcut list window, open the shortcut list window, select the shortcut, and choose **Shortcuts > Edit Shortcut** .

For more information about working with shortcut list windows, see the [shortcut list windows in-depth](#) help topic. Also see the [shortcuts](#) topic for more general information about shortcuts.

## [Fetch Help](#) > [Windows](#) > **Shortcut List Windows**

A shortcut list window collects a group of [shortcuts](#) in a single window. This list of shortcuts can be saved and opened again later.

This is a detailed description of shortcut list windows; for a quick guide, see the [shortcut list window introduction](#) help topic.



Name	Hostname	Security	Username	Path
 Fetch Softworks	ftp.fetchsoftworks.com			
 My web page	webserver.net		joesmith	www
 Project status	fetchsoftworks.com		jsmith12	projects/status.doc

The [Fetch Shortcuts window](#), which Fetch saves and opens automatically, is a shortcut list window. Other shortcut list windows are not saved or reopened automatically, and their shortcuts do not appear in the [New Connection dialog](#) or the **Shortcuts** menu. If you expect to use a shortcut often, you should add it to the Fetch Shortcuts window, not a separate shortcut list window.

To create a shortcut list window, choose **Shortcuts > New Shortcut List**. You do not need to create the Fetch Shortcuts window, it is created automatically the first time you start Fetch.

To open a shortcut in a shortcut list window, double-click it, or select it and press Command-Down arrow. This will either open the remote folder or start downloading the file it refers to. You can also drag a shortcut to a Finder window or the desktop to start a download. Command-double-click a shortcut to open it in a new transfer window, even if a window to that server is already open.

Uploads can be started by dragging files or folders to a shortcut for the destination. Fetch will automatically open a connection to the location referenced by the shortcut and start the upload.

In a shortcut list window, the three kinds of shortcuts are represented with different icons. Folder or server shortcuts have a folder icon. File shortcuts have a blank document icon. Unknown shortcuts (which may refer to a folder or file) have a combined folder/document icon.

Shortcut list windows have columns listing each shortcut's name, server hostname, security indicator icon (the padlock icon), the username associated with the shortcut (if any), and the path the shortcut refers to (if any). You can sort the shortcuts by clicking any column heading; click the column heading a second time to reverse the sort order.

The security icon indicator will be one of the following: a locked padlock for secure, encrypted connections (SFTP, FTP with TLS/SSL, Kerberos with encryption); an unlocked padlock for connections where your password is sent securely but data transfers won't be encrypted (FTP with TLS/SSL or Kerberos without encryption); and no icon for connections with no security (regular FTP connections). See the [padlock icon](#) and [security](#) help topics for more information.

You can add shortcuts to a shortcut list window in a number of ways:

- The **Shortcuts > New Shortcut** command lets you create a shortcut manually in the active shortcut list window (or in the Fetch Shortcuts window if none are active)
- The optional **New Shortcut** button in the transfer window toolbar also lets you create a

shortcut in the Fetch Shortcuts window manually.

- You can drag files or folders from a transfer window into a shortcut list window to create shortcuts for those items.
- You can drag shortcuts from one shortcut list window into another to copy them to the new window.
- You can cut or copy shortcuts from one window and paste them into another.
- You can paste Fetch addresses (FTP URLs) into a shortcut list window.
- You can drag text Fetch addresses (FTP URLs), for example from an email program or newsreader, into a shortcut list window.
- If you've set up [WebView](#) for a server, you can also paste or drag web addresses (HTTP URLs) for that server into a shortcut list window.

To remove a shortcut from a shortcut list window, select the shortcut and then choose **Edit > Delete**, press the Delete key, or drag the shortcut to the Mac OS Trash icon.

The entire shortcut list can be saved as a [shortcut list document](#) for future use with **File > Save** or **File > Save As**, and reopened with **File > Open** .

To edit a shortcut in a shortcut list window, open the shortcut list window, select the shortcut, and choose **Shortcuts > Edit Shortcut** . You can also Option-Shift-double-click the shortcut to open the [Edit Shortcut dialog](#).

You can also edit the shortcut's information directly in the shortcut list window by clicking once to select the shortcut, and then clicking again on the piece of information you want to edit. An edit box appears around the text. Edit the text, and press Return to confirm your changes.

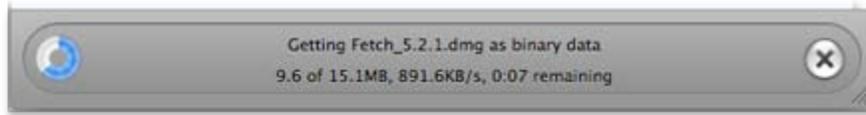
Option-double-clicking a shortcut in a shortcut list window brings up the New Connection dialog, filled out with information from the shortcut. You can also do this from the keyboard by selecting a shortcut and pressing Command-Option-Down arrow.

You can access commonly used commands by Control-clicking a shortcut to display the [shortcut list window contextual menu](#). In particular, the contextual menu is the only way to access the [Duplicate](#) command, which will make a copy of a shortcut in the same window.

Also see the [shortcuts](#) topic for more general information about shortcuts.

## [Fetch Help](#) > [Windows](#) > **Status Pane**

The area at the bottom of the [transfer window](#) is the status pane. It contains information about what Fetch is doing at any given time.



If a file transfer is in progress, on the left of the status pane you'll see a donut-shaped progress indicator showing how much of the current file has been transferred. The status pane also displays the name of the file currently being transferred, the download mode or upload format being used, the file's approximate size, the amount of data that's been transferred so far, the average speed of the transfer, and an estimate of how long it will take for the transfer to complete.

When a transfer or other action is in progress, you can click the **Stop**  button on the right side of the status pane to halt the transfer or action. You can also stop the current action by typing Command-**.** (Command-period). Once you have done so, the button will disappear and the status pane will display "Stopping..."

Most servers support the stop command and canceling will only take a few seconds. Some servers, however, ignore the stop command and insist on completing the current action. In these cases, the status line will display "Stopping..." for quite a while. The only ways to quickly exit from this situation are to close the connection by closing the transfer window, or to quit Fetch.

The status pane may also give other messages telling you what Fetch is doing when an action is not instantaneous, such as "Connecting..." (establishing a connection), "Changing folder name..." (when you're moving to another remote folder), "Getting file list..." (reading the contents a remote folder), and so forth.

When no action is taking place, the status pane tells you the status of your connection to the server. When the status pane displays "Connected", actions happen without Fetch having to reestablish the connection, whereas when it displays "Disconnected", Fetch will have to reestablish the connection before performing an action. It's normal for servers to become disconnected after your connection has been inactive for a few minutes, and Fetch almost always reestablishes connections transparently, so you don't need to worry if your transfer window is connected or disconnected.



## [Fetch Help](#) > [Windows](#) > Text Windows

Text windows are used any time Fetch needs to display text-based information. They behave similarly to any other Macintosh window that shows text, although you cannot edit the contents of Fetch text windows.

Text windows include the [Fetch Transcript](#) window, any text file which you've displayed using the [View](#) command (on Mac OS X 10.4), [server message windows](#), and the [verbose file list](#) window.

In any text window, you can:

- Select text using the mouse and keyboard, and copy it to the clipboard.
- Select and drag text to any location that accepts drag-and-drop, such as a window in another application or the Finder desktop.
- Search for words or phrases in text windows with the [Find](#) command.
- Print the contents of the window with the [Print](#) command.
- Save the contents of the window as a text file with the [Save a Copy As](#) command.
- Change the font and size of the text with the [Set Font and Size](#) command.
- Change the tab width used to display tabs with the [Set Font and Size](#) command.

You can edit a remote text file on a server using the [Edit](#) command. If you want to make changes to a text file locally but not on the server, use the [Save a Copy As](#) command to save it to disk and then edit it with your favorite text editor.

## [Fetch Help](#) > [Windows](#) > Transfer Window

The transfer window lets you copy files between your Macintosh and a server, as well as display or modify files on a server. The Fetch transfer window appears once you have successfully opened a connection to a server.



Fetch creates a new transfer window for each connection you open from the [New Connection dialog](#) or with a [shortcut](#).

The window title bar shows the name of the current folder, and then either the name of the shortcut matching this server or the server's hostname.

At the top of the transfer window is the [transfer window toolbar](#), which contains buttons for common commands. You can modify the buttons shown in the toolbar, or hide it altogether.

Click the **Path** button in the toolbar, or Command-click the title, to change to any folder that contains the current folder. Use the **Back** button in the toolbar to return to the folder you were previously viewing.

Just below the toolbar is the info bar, which shows the number of items in the current folder, if the list is cached, the [Refresh](#)  button, and a [padlock icon](#) indicating the security of your connection.

In the middle of the window is the [file list](#), which shows the remote files and folders on the server. Click a column heading to sort the file list by that column; click the column heading a second time to reverse the sort order.

Click items in the file list to select them. Then choose a command from the toolbar, or from the menus, to do something with those items.

Double-click files in the file list to get (download) the files, or double-click a folder to show the contents of that folder in the file list.

Drag files or folders from the Mac OS X Finder to the file list to upload those items to the server. You can also use the **Put** button in the toolbar to select items for upload if they are not easily accessible for dragging.

Rename files by clicking once to select an item, and then pressing the Return key to display the edit box. Type a new name or edit the existing name, and press Return again when you're

done.

At the bottom of the window is the [status pane](#), which displays information about the current transfer or action, such as which file you are downloading and how much more remains to be downloaded. You can stop a transfer or other lengthy action with the **Stop**  button.

For more in-depth information about the transfer window, see the following topics:

- [Transfer window toolbar](#)
- [File list](#)
- [Status pane](#)

 [Fetch Help](#) > [Windows](#) > **Verify Certificate Dialog**

When you connect to a server using [FTP with TLS/SSL](#), the server sends Fetch a certificate that Fetch uses to verify the security and identity of the server.

A server's certificate is issued by a certificate authority, which also vouches for the legitimacy of the certificate by digitally signing it. Your Macintosh has a list of certificate authorities that it trusts, and if the server's certificate is signed by one of those authorities, it will be accepted automatically.

However, if the certificate is expired, or not signed by a trusted authority, or has another problem, Fetch will display the Verify Certificate dialog when you try to connect.

The Verify Certificate dialog shows you the name of the server you are connecting to, a description of why Fetch is displaying the warning, and an opportunity to examine the certificate in more detail, so that you can decide whether you want to continue connecting to the server or not. You can also choose to have this certificate always accepted in the future, so that Fetch will not warn you about it again.

While you may encounter servers where the certificate warning is not a serious concern, if you have any doubts about whether you should continue connecting or not, click **Cancel** and contact the server administrator, the hosting company who provides the server, or the person who gave you access to the server, to make sure that the server's security is not compromised, and that you are not connecting to the wrong server.

Click the **Show Certificate** button to see the standard Mac OS X certificate information display, which shows the name of the server the certificate was issued for, which authority issued it, when it expires, and cryptographic information about the certificate.

Clicking the **Show Certificate** button will also display the "Always trust these certificates" checkbox. If you check this box and then click the **Continue** button, this certificate will be added to your keychain, so that you will not be warned when you connect to this server in the future. Be very sure you want to accept these certificates as valid before checking this box.

Click the **Cancel** button to stop connecting to this server.

Click the **Continue** button to connect to the server despite the warning. If you checked the "Always trust these certificates" checkbox, this certificate will be added to your keychain and your list of trusted certificates.

Even if you do not check the "Always trust these certificates" checkbox, once you connect to a server, Fetch will not display the certificate warning for that server again for as long as Fetch is running.

For more information about certificates, see the [Mac OS X help topics on certificates](#).

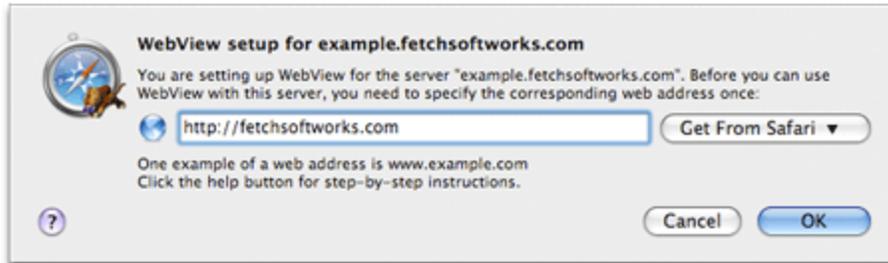
Related topics:

- [FTP with TLS/SSL](#)

## [Fetch Help](#) > [Windows](#) > **WebView Dialog (Web Address Only)**

WebView lets you set up a correspondence between files you are viewing in Fetch and the address of the webpage they are part of, so that you can easily review changes to your site in a web browser and copy the web addresses of those files.

The first time you choose **WebView** for a server, Fetch displays the WebView dialog requesting the web address of your server.



Before you can use WebView, you must tell Fetch the web address of your server. The WebView dialog will only appear the first time you choose WebView; once WebView is set up for a server, you do not need to enter this information again.

The next time you choose **WebView** when connected to the same server, Fetch will show the selected files or folder in your web browser instead of the WebView dialog. You can also use the [Copy Web Address](#) command to copy a web address (also called a link or URL) of the selected items in Fetch. For more information about WebView, see the [WebView](#) help topic.

To set up WebView for a folder using the WebView dialog:

1. Look at the server and folder name mentioned in the dialog, or displayed in the window's title bar. You are setting up WebView for that folder. If it is not your home folder or the folder containing your webpages, you may want to cancel and go to one of those folders, and then set up WebView.
2. Enter the web address corresponding to this folder and server in the edit field.
  - a. Usually you should enter your domain name, or the address for your website's homepage.
  - b. Examples of web addresses are: `http://www.example.com/myusername/` or `www.ExampleDomainName.com`
  - c. You can use the **Get From Safari** pop-up menu to get the web address of a webpage currently open in Safari.
3. Click the **OK** button. Fetch should show the selected file or folder in your web browser.

If the webpage that opens after you have entered a web address is incorrect — that is, it doesn't show you the page you expected, or it shows you an error — you may have entered an incorrect web address and need to try again.

To edit a WebView entry, either hold down the Option key and click the **WebView** button, or use the [WebView Preferences](#) pane.

The **Get From Safari** pop-up menu is only available when Safari is running and has open webpages. While WebView works with any web browser, the WebView dialog can only show you

a list of open webpages from Safari due to limitations of other browsers.

Why doesn't Fetch automatically figure out the web address for you? Unfortunately, there's no standard for specifying the web address of files on a server; each server uses its own method, and the method used by one server may not apply on another. There is no way for Fetch to ask your server which method it uses, so Fetch has to ask you instead. While the information for some popular servers is built into Fetch, in most cases Fetch needs your input. If you are not sure what the corresponding web address should be, you should contact your hosting provider or server administrator.

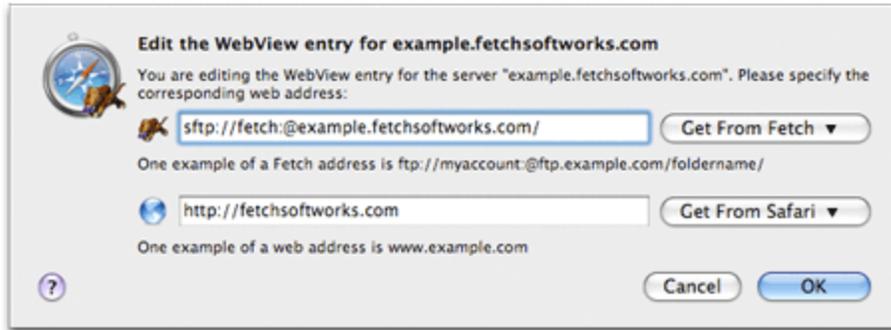
Related topics:

- [WebView](#)
- [WebView Preferences](#)
- [WebView command](#)
- [WebView dialog \(with Fetch address\)](#)
- [Copy Web Address](#)

## [Fetch Help](#) > [Windows](#) > **WebView Dialog (With Fetch Address Field)**

WebView lets you set up a correspondence between files you are viewing in Fetch and the address of the webpage they are part of, so that you can easily review changes to your site in a web browser and copy the web addresses of those files.

When you click the Add or Edit buttons in the [WebView Preferences](#) pane, Fetch displays the extended WebView dialog so you can create a new WebView entry or modify the Fetch and web addresses of an existing entry.



Before you can use WebView, you must set up a correspondence between a folder in Fetch and the address of a webpage that folder represents. Normally, the best way to set up WebView is to connect to your file server, and then click the **WebView** button in the toolbar. (See the [setting up WebView](#) topic for more detailed instructions.) When you do that, you only need to enter the web address, because Fetch can get the Fetch address of your server from your connection. When you click the Add (+) button in the WebView Preferences pane, you must enter both pieces of information yourself.

Once you've set up WebView for a server, you can then click the **WebView** button in the toolbar of transfer windows to show the selected files or folders in your web browser. You can also use the [Copy Web Address](#) command to copy a web address (also called a link or URL) of the selected items in Fetch. For more information about WebView, see the [WebView](#) help topic.

To set up WebView for a server using the extended WebView dialog:

1. In the first field (with a Fetch icon), enter the Fetch address of the folder and server you want to set up WebView for. The Fetch address must contain your username, the server's hostname, and the path to your home folder on the server.
  - a. An example of a Fetch address is  
`ftp://myaccount:@ftp.example.com/foldername/`
  - b. Fetch addresses must be URLs — see the [Uniform Resource Locators \(URLs\)](#) help topic for more information.
  - c. Usually you should enter the address of your home folder, or the folder containing your webpages.
  - d. You can use the **Get From Fetch** pop-up menu to get the Fetch address of a currently open Fetch transfer window.
2. In the second field (with a globe icon), enter the web address corresponding to the Fetch address.
  - a. Usually you should enter your domain name, or the address for your website's homepage.
  - b. Examples of web addresses are: `http://www.example.com/myusername/` or `www.ExampleDomainName.com`

- c. You can use the **Get From Safari** pop-up menu to get the web address of a webpage currently open in Safari.
3. Click the **Add** button.

To test your WebView setup, you will need to connect to your server, possibly select a file, and click the **WebView** button. If the webpage that opens is incorrect — that is, it doesn't show you the page you expected, or it shows you an error — you may have entered an incorrect Fetch or web address and need to edit your entry.

To edit a WebView entry, either hold down the Option key when clicking the **WebView** button, or use the [WebView Preferences](#) pane.

When you create a WebView entry, Fetch tries to generalize it — if Fetch notices that the same folder name is at the end of both the Fetch address and web address, it will remove that folder name from both addresses to create a WebView entry that will apply to other folders on the server as well. While this should almost always be what you want, if for some reason you need to have the same folder name at the end of each address, you can edit the WebView entry using the WebView Preferences pane, and Fetch will not remove matching folder names.

The **Get From Fetch** pop-up menu is only available when there are open [transfer windows](#).

The **Get From Safari** pop-up menu is only available when Safari is running and has open webpages. While WebView works with any web browser, the WebView dialog can only show you a list of open webpages from Safari due to limitations of other browsers.

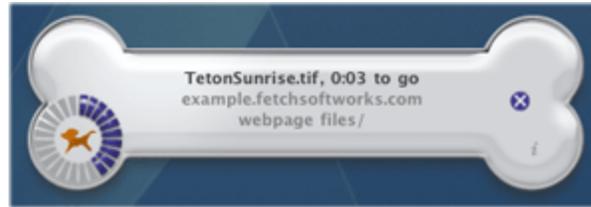
Why doesn't Fetch automatically figure out web addresses for you? Unfortunately, there's no standard for specifying the web address of files on a server; each server uses its own method, and the method used by one server may not apply on another. There is no way for Fetch to ask your server which method it uses, so Fetch has to ask you instead. While the information for some popular servers is built into Fetch, in most cases Fetch needs your input. If you are not sure what the corresponding web address should be, you should contact your hosting provider or server administrator.

Related topics:

- [WebView](#)
- [WebView Preferences](#)
- [Setting up WebView](#)
- [WebView dialog \(web address only\)](#)
- [WebView command](#)
- [Copy Web Address](#)

## [Fetch Help](#) > Fetch Widget

The Fetch Dashboard widget lets you easily upload files and folders to a specific server by dragging and dropping items on the widget. After the upload has started, the Fetch widget displays the progress of the transfer.



Dashboard is a feature of Mac OS X that provides a convenient way to access mini-applications and utilities called "widgets." If you are not familiar with Dashboard, you may want to read the [Mac OS X help topics on Dashboard and widgets](#).

When you first open Fetch, you will be asked if you want to install the Fetch widget. If you choose not to install it at that time, you can always install it later by choosing **Fetch** > [Install Dashboard Widget](#).

If you choose to install the widget when asked, Fetch will open Dashboard and display a new copy of the Fetch widget. If you've set a [default shortcut](#) in Fetch, the widget is automatically configured with that shortcut; otherwise, the widget starts unconfigured.

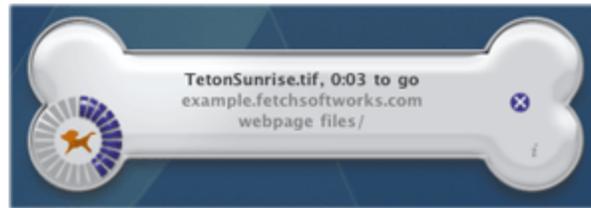
Using the Fetch widget is simple:

1. Click the info (i) button to flip the widget to its back.
2. Choose a recent connection or a shortcut as the widget's destination by clicking one of the two pop-up buttons. If the destination you want is not listed, you must [create a shortcut](#) in the Fetch application first, and then return to Dashboard.
3. Click the **Done** button to return to the front of the widget.
4. In the Finder, select the files and folders you want to upload.
5. Click one of the items and start dragging.
6. With your other hand, press the Dashboard keyboard shortcut (usually the F12 key) to open Dashboard.
7. Continue dragging the files until they are over the Fetch widget, and drop the files by releasing the mouse button.
8. The widget will open a connection in Fetch and start uploading the files.

For more detailed steps on configuring and using the widget, see the [Using the Fetch widget for uploading](#) help topic.

The Fetch widget, like most widgets, has two sides, a front and a back.

On the front of the widget you'll find:

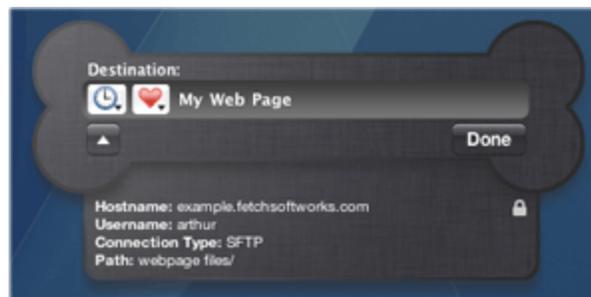


- the status line — displays the progress of the current transfer, or the final status of the last transfer;
- the hostname and path lines — list the widget's destination;
- the donut-shaped progress indicator — shows how much of the current file has been transferred;
- the Open Connection button (when no transfer is in progress) — when clicked, dismisses Dashboard and opens a connection to the widget's destination in the Fetch application;
- the Stop button (when a transfer is in progress) — when clicked, stops the current transfer;
- the info (*i*) button — when clicked, flips the widget to its back so you can configure it.

If the attempt to upload the items fails, the widget will display a More Details button. Click the More Details button to dismiss Dashboard and go to the Fetch application to see an explanation of why the transfer failed.

If Fetch requires a password to open the connection, the status line will report "Waiting for password..." and display an arrow button. Click the arrow button to dismiss Dashboard and go to the Fetch application to enter the password. The upload will then begin.

On the back of the widget you'll find:



- the Recent Connections button — click to configure the widget with a recent connection;
- the Shortcuts button — click to configure the widget with one of your Fetch shortcuts;
- the name of the recent connection or shortcut you've chosen as the widget's destination;
- the disclosure button to display the info drawer — click to display or hide detailed information about the widget's destination;
- the info drawer (optional) — shows the hostname, username, connection type, path, and [security](#) of the widget's destination;
- the Done button — click to return to the widget's front when you're done choosing a destination.

Note that the widget will not list shortcuts to files as choices for configuring the widget, since you cannot upload to a file. If the destination you want is not listed, you must [create a shortcut](#) to it in the Fetch application first, and then return to Dashboard.

When you upload items with the widget, they are uploaded using the [default upload format](#), unless a transfer window to the destination is already open and you have changed the upload format for that transfer window to something other than the default.

You can open as many copies of the Fetch widget as you want, and configure each one with a different shortcut or recent connection.

To open a new copy of the Fetch widget:

1. Open Dashboard, either by clicking the Dashboard icon in the Dock or by pressing the Dashboard keyboard shortcut (usually the F12 key).
2. Click the Open (+) button in the bottom-left of the screen to reveal the widget bar.
3. Use the arrow buttons in the widget bar to find the icon for the Fetch widget if necessary.
4. Click the Fetch widget icon or drag it out of the widget bar to open a new Fetch widget.
5. Click the Close (x) button to hide the widget bar again.

To close a copy of the Fetch widget, hold down the Option key and move your mouse over the copy of the widget you want to close. A close (x) button will appear in the upper left corner of the widget. Click the close button.

For information on uninstalling the Fetch widget, see the [Uninstalling Fetch](#) help topic.

To change the keyboard shortcut for displaying Dashboard, open the **Dashboard & Exposé** pane in the **System Preferences**. (On Macintosh laptops that use the F12 key to eject a disc, Fn-F12 should open Dashboard, but you may want to change it to something more convenient.)

The following topics offer in-depth information about the commands and capabilities of Fetch. Select a topic for more information about it. If you don't see what you're looking for, also check the [Using Fetch](#) list of topics, or use the Search field above.

[About Fetch](#)

[Back](#)

[Check for Update](#)

[Clear Menu \(Go menu\)](#)

[Clone Window](#)

[Close](#)

[Copy](#)

[Copy Fetch Address](#)

[Copy Web Address](#)

[Cut](#)

[Delete](#)

[Delete Item](#)

[Duplicate](#)

[Edit](#)

[Edit New Text File](#)

[Edit Other](#)

[Edit Shortcut](#)

[Edit With](#)

[Edit With submenu](#)

[Enable Encryption](#)

[Enter Serial Number](#)

[Find / Find Again](#)

[Format submenu](#)

[Get](#)

[Get As](#)

[Get Info](#)

[Go to Folder](#)

[Home](#)

[Install Dashboard Widget](#)

[List of shortcuts](#)

[Make Shortcut](#)

[Mirror](#)

[Mode submenu](#)

[New Connection](#)

[New Folder](#)

[New Shortcut](#)

[New Shortcut List](#)

[Open](#)

[Open Folder](#)  
[Page Setup](#)  
[Parent Folder](#)  
[Paste](#)  
[Path](#)  
[Preferences](#)  
[Print](#)  
[Purchase](#)  
[Purchase Upgrade](#)  
[Put](#)  
[Quick Look](#)  
[Quit](#)  
[Recent Connections](#)  
[Recently visited folders](#)  
[Refresh](#)  
[Resume Download](#)  
[Save](#)  
[Save As](#)  
[Save Droplet Shortcut](#)  
[Save Shortcut](#)  
[Save a Copy As](#)  
[Search](#)  
[Send FTP Command](#)  
[Set Font and Size](#)  
[Set Upload Permissions](#)  
[Stop](#)  
[View](#)  
[View As](#)  
[View as Text](#)  
[View Other](#)  
[View Verbose File List](#)  
[WebView](#)

 [Fetch Help](#) > [Commands](#) > **About Fetch**

The **About Fetch** command displays a window containing information about your copy of Fetch.

This command is available as:

- The **About Fetch** command in the [Fetch menu](#)

The About Fetch window displays the version number of your copy of Fetch, your Fetch serial number and the name associated with the serial number (or the expiration date of your trial period if you haven't registered yet), and contact information for Fetch Softworks. You can also view the complete credits for Fetch, create an email to Fetch support, visit the Fetch Softworks website, and [check for updates](#) to Fetch.

The **Back** command returns you to the remote folder you were viewing before the current one, and updates the file list.

This command is available:

- As the **Back** command in the [Go menu](#)
- As the **Back** button in the [transfer window toolbar](#)
- By swiping left on laptop trackpads that support gestures

If you haven't viewed any other folders besides the current one, the **Back** command is disabled.

See the [changing folders](#) help topic for other ways to move between folders.

## [Fetch Help](#) > [Commands](#) > **Check for Update**

The **Check for Update** command will check to see if you're using the newest version of Fetch, and provides an easy way to download updates if you're not.

This command is available as:

- The **Check for Update** command in the [Fetch menu](#)
- The **Check for Update** button in the [About Fetch](#) window

You must have an active Internet connection to use the **Check for Update** command.

The **Check for Update** command contacts the Fetch Softworks servers over the Internet, and compares the version you are using with the latest available version. If the latest version is newer than the version you are using, Fetch will display the [Fetch Update](#) dialog so you can get more information about the update, and download it and install it if you want.

In some cases, there may be two updates available for Fetch at the same time — a free update and an update that requires a purchase. When two updates are available, Fetch will show the [Fetch Update \(Two Updates\)](#) dialog instead.

If you have limited access to the Macintosh you are using, you may receive the message "You do not have permission to update Fetch" when you choose the **Check for Update** command. Only the administrator of the Macintosh can update Fetch.

Normally Fetch checks for updates and notifies you of them automatically, so you don't need to use the **Check for Update** command. But if you have disabled the automatic update check, or if you skipped over or delayed installing an update, you can use the manual **Check for Update** command to see if you're up to date.

Related topics:

- **Automatically check for updates** preference in the [Miscellaneous Preferences](#) pane
- [Software Update](#)

 [Fetch Help](#) > [Commands](#) > **Clear Menu (Go menu)**

The **Clear Menu** command at the bottom of the **Go** menu clears out the list of [recently visited folders](#) that is shown in the **Go** menu and when you click the **Recent** button in the transfer window toolbar.

This command is available as:

- The **Clear Menu** command in the [Go menu](#)

There is also a **Clear Menu** command in the [Recent Connections](#) submenu; that command clears out the recent connections and does not affect the list of recently visited folders.

Related commands:

- [Recently visited folders](#)

The **Clone Window** command lets you create a second connection to a server by duplicating a transfer window.

This command is available as:

- The **Clone Window** command in the [file list contextual menu](#)

Control-click in any [transfer window](#) and choose **Clone Window** from the contextual menu to open a new transfer window that is connected to the same server and folder as the original transfer window.

Note that when you have two connections open to the same server, changes you make in one transfer window may not be reflected in the other transfer window until you use the [Refresh](#) command.

Normally Fetch tries not to open duplicate connections to the same server — for example, when you open a shortcut from the Fetch Shortcuts window and a connection to that server is already open, it does not open a new transfer window. The **Clone Window** command may be useful when you do want to open the same connection twice.

You can also open a second, new transfer window to a server by Command-double-clicking a remote folder in a file list or Command-double-clicking a shortcut. Another way is to use the [New Connection](#) command to open a second, new connection to a server.

 [Fetch Help](#) > [Commands](#) > **Close**

The **Close** command closes the active window.

This command is available as:

- The **Close** command in the [File menu](#)

You can also close a window by clicking the window's close button.

If you hold down the Option key while choosing **Close**, the command changes to **Close All** and all open Fetch windows will be closed.

If you try to close a [transfer window](#) where a transfer is in progress, you will be warned that the connection is active before the window is closed.

If you are closing a [shortcut list window](#) and have unsaved changes, Fetch will ask if you want to save them before closing the window.

## [Fetch Help](#) > [Commands](#) > **Copy / Copy With Fetch Address**

The **Copy** command lets you copy text window contents, or copy and then paste shortcuts in shortcut list windows.

This command is available as:

- The **Copy** command in the [Edit menu](#)
- The **Copy** command in the [shortcuts list window contextual menu](#)

You can use the **Copy** command to copy text from any place you can select text.

In [shortcut list windows](#), you can select shortcuts and copy them to be pasted in other shortcut list windows.

If you copy a shortcut and paste into a window that accepts text, either a web address or a Fetch address (URL) equivalent to that shortcut will be pasted, depending on whether you have configured [WebView](#) for the server referred to by the shortcut. To copy a Fetch address when WebView is configured for the shortcut, hold down the Option key when choosing **Copy**, and the command will become **Copy With Fetch Address**.

If you select an item in a [file list](#), the **Copy** command becomes the [Copy Fetch Address](#) or [Copy Web Address](#) command and copies a [Uniform Resource Locator \(URL\)](#) to that item on the clipboard. See those commands' help topics for more information.

To copy a remote file or folder from one server to another, drag the remote file or folder from one [transfer window](#) to another.

You can use the **Copy** and **Paste** commands to upload files and folders in Fetch, or to copy files and folders from one folder to another on the same server, or from one server to another. See the [uploading using copy and paste](#) or [copying files using copy and paste](#) help topics for more information.

You can also use the [Duplicate](#) command to make another copy of a remote file on a server.

Related topics:

- [Copy Fetch Address](#)
- [Copy Web Address](#)
- [Paste](#)
- [Cut](#)
- [Duplicate](#)

## [Fetch Help](#) > [Commands](#) > **Copy Fetch Address**

The **Copy Fetch Address** command copies to the clipboard Fetch addresses (FTP [URLs](#)) referring to the files or folders selected in the [transfer window](#).

This command is available as:

- The **Copy Fetch Address** command in the [Edit menu](#), when a transfer window is active
- The **Copy Fetch Address** command in the [file list contextual menu](#)

When no files or folders are selected, the **Copy Fetch Address** command copies an FTP URL to the current folder displayed in the transfer window, instead of any of the items in it.

**Copy Fetch Address** is only directly available when you haven't set up [WebView](#) for the server you are connected to. Otherwise, the default Copy command will be [Copy Web Address](#). You can access the **Copy Fetch Address** command instead by holding down the Option key when choosing **Copy Web Address**.

You can use the **Copy** and **Paste** commands to upload files and folders in Fetch, or to copy files and folders from one folder to another on the same server, or from one server to another. See the [uploading using copy and paste](#) or [copying files using copy and paste](#) help topics for more information.

Related topics:

- [Copy Web Address](#)
- [Copy](#)
- [Uniform Resource Locators \(URLs\)](#)



## [Fetch Help](#) > [Commands](#) > **Copy Web Address**

The **Copy Web Address** command copies to the clipboard a web addresses (HTTP [URLs](#)) referring to the files or folders selected in the [transfer window](#).

This command is available as:

- The **Copy Web Address** command in the [Edit menu](#), when a transfer window is active and WebView has been set up for that server
- The **Copy Web Address** command in the [file list contextual menu](#), when WebView has been set up for that server

When no files or folders are selected, the **Copy Web Address** command copies a web address (HTTP URL) to the current folder displayed in the transfer window, instead of any of the items in it.

The **Copy Web Address** command is only available when you have set up [WebView](#) for the server you are connected to. If you have not set up WebView for that server, only the [Copy Fetch Address](#) command will be available.

When you have set up WebView for a server, the default Copy command is **Copy Web Address**. You can access the **Copy Fetch Address** command instead by holding down the Option key when choosing **Copy Web Address**.

You can use the **Copy** and **Paste** commands to upload files and folders in Fetch, or to copy files and folders from one folder to another on the same server, or from one server to another. See the [uploading using copy and paste](#) or [copying files using copy and paste](#) help topics for more information.

Related topics:

- [WebView](#)
- [Setting up WebView](#)
- [Copy Fetch Address](#)
- [Copy](#)
- [Uniform Resource Locators \(URLs\)](#)

The **Cut** command lets you cut text or shortcuts.

This command is available as:

- The **Cut** command in the [Edit menu](#)
- The **Cut** command in the [shortcuts list window contextual menu](#)

You can use the **Cut** command to cut text from any place you can edit text.

In [shortcut list windows](#), you can select shortcuts and cut them to be pasted in other shortcut list windows. If you cut a shortcut and paste it into a window that accepts text, a Fetch address (URL) equivalent to that shortcut will be pasted.

Related topics:

- [Paste](#)
- [Copy](#)
- [Duplicate](#)

 [Fetch Help](#) > [Commands](#) > **Delete**

The **Delete** command lets you delete shortcuts from a shortcut list window, remove entries from the [Resume Download](#) window, or delete text in an edit field.

This command is available as:

- The **Delete** command in the [Edit menu](#)
- The **Delete** command in the [shortcut list window contextual menu](#)

To delete shortcuts, select them in the [Fetch Shortcuts](#) window or another [shortcut list window](#) and choose the **Delete** command. You can also delete shortcuts by selecting them and pressing the Delete key, or by dragging them to the Mac OS Trash icon.

You will be asked for confirmation before any shortcuts are deleted. Deleting shortcuts cannot be undone. Note that **Delete** only removes the shortcuts; it does not delete any of the remote items the shortcut refers to.

If you have selected multiple shortcuts and give confirmation, all the shortcuts will be deleted.

To remove entries in the Resume Download window, select the entry and choose the **Delete** command, or press the Delete key.

To delete remote files or folders, or to delete the remote files or folders referred to by shortcuts, use the [Delete Item](#) command in the **Remote** menu instead.



The **Delete Item** command lets you delete files and folders on a server.

This command is available as:

- The **Delete Item** command in the [Remote menu](#)
- The **Delete** button in the [transfer window toolbar](#)
- The **Delete Item** command in the [file list contextual menu](#)

To delete remote files or folders, select them in a [file list](#) and choose the **Delete Item** command, or click the **Delete** button in the toolbar. You can also delete remote items by selecting them and typing Command-Delete, or by dragging them to the Mac OS Trash icon.

You will be asked for confirmation before any items are deleted. Deleting remote items cannot be undone.

If you select multiple remote items and give confirmation, all the selected items will be deleted.

If you select a remote folder and choose this command, Fetch will delete the folder's contents before deleting the folder itself.

If you hold down the Shift key while choosing this command, the command changes to **Delete Other Item** and Fetch will prompt you for the name of a file or folder to delete, and then that item will be deleted. **Delete Other Item** is only available in the main menu when a transfer window is the active window.

You must have the necessary privileges to delete remote items. Otherwise, you will get an error message and the items will not be deleted.

To delete shortcuts, use the [Delete](#) command in the **Edit** menu instead.

## [Fetch Help](#) > [Commands](#) > **Duplicate**

The **Duplicate** command either duplicates files on a server, or duplicates a shortcut in Fetch Shortcuts or another shortcut list window.

This command is available as:

- The **Duplicate** command in the [file list contextual menu](#)
- The **Duplicate** command in the [shortcut list window contextual menu](#)

When one or more files are selected in a file list, the **Duplicate** command lets you duplicate them. It does this by using a [server-to-server transfer](#). Fetch opens a second connection and transfer window to the server, and transfers the file from the server to itself, resulting in a new copy of the file. The name for the new file is chosen by the server, so while it should be a variant of the original filename (such as "myfile.1" for a duplicate of "myfile"), some servers will assign a meaningless name.

When a shortcut is selected in [Fetch Shortcuts](#) or another [shortcut list window](#), the **Duplicate** command makes a copy of that shortcut in the same window. You can then [edit the new shortcut](#) to make any changes. This is handy if you want to make a set of very similar shortcuts.

Related topics:

- [Moving files and folders](#)
- [Copying files using copy and paste](#)

The **Edit** command lets you make changes to any kind of file in another application and save the changes back to the server automatically.

This command is available as:

- The **Edit** command in the [Remote menu](#)
- The **Edit** button in the [transfer window toolbar](#)
- The **Edit** command in the [file list contextual menu](#)
- The **Edit** command in the [shortcut list window contextual menu](#)

When you select a file in the file list and choose the **Edit** command, Fetch downloads the file to a temporary location and opens it in the editor application for that kind of file.

After you use the editor's **Save** command to save your changes, Fetch automatically uploads the changed file back to its original location. (A copy of the file will not be left on your Macintosh.) This provides a convenient way to edit files on a server.

Fetch supports editing any kind of file with any application on your Macintosh. The default editor for a kind of file is usually the same as the application that would normally open that kind of file, although in certain cases the editor may be different from the normal opening application because that application cannot edit files.

If Fetch does not open the application you would like to use for editing a certain kind of file — or you would like to use a different editor temporarily — hold down the Option key when choosing the **Edit** menu item or clicking the **Edit** button and the command will become [Edit With](#). Fetch will ask you to choose a different application to use for editing and whether you want Fetch to remember this choice for the future or not.

You can also choose a different editor to use temporarily with the [Edit With submenu](#) in the **Remote** menu or the contextual menus.

You can also [change the editor application](#) for a specific kind of file in the [Transfer Options](#) pane of the [info window](#).

If the editor for the kind of file you've chosen is not available, or if Fetch cannot determine what kind of file you've selected, Fetch will ask you which editor you want to use.

You must leave Fetch open while you edit files in order for them to be saved back to the server (you do not have to remain connected to the server, however).

If no file is selected, the **Edit** command changes to [Edit New Text File](#), which you can use to create new text files which will be automatically uploaded to the server when you save them. Fetch will prompt you for the name of the text file to be created and then open the appropriate text editor.

If you hold down the Shift key when selecting the **Edit** command, the command becomes [Edit Other](#) and Fetch will prompt you for a filename. The file must already exist on the server. **Edit Other** is only available when a transfer window is the active window.

You can tell Fetch to automatically edit files when you double-click them. See the [editing files](#)

[automatically instead of downloading](#) help topic.

If you want Fetch to open your files in your favorite application after downloading, but not save them back to the server automatically, see the [opening files automatically after downloading](#) help topic.

When you use the **Edit** command to edit a file with BBEdit 9.2 or later (or TextWrangler 3.0 or later), Fetch will pass along encoding specified by the **Preferred encoding** preference as the encoding the file should be displayed in. However, if your file already specifies its encoding with a BOM or an HTML meta tag, BBEdit should detect that and use the proper encoding regardless of Fetch's **Preferred encoding** setting. Again, the contents of the file are not changed; if the encoding sent to BBEdit is not the correct one, you can use BBEdit's **Reopen Using Encoding** command to display it in the desired encoding. Currently, no other text editors support receiving the preferred encoding setting from Fetch, but they may be able to detect a file's encoding if it is specified by the file itself. For more information about encodings, see the [Character Encoding & Translation](#) help topic.

Related commands:

- [Quick Look](#) (or [View](#) on Mac OS X 10.4)



The **Edit New Text File** command is a special version of the [Edit](#) command that lets you create and edit a new text file on a server.

The **Edit New Text File** command is available:

- In place of the **Edit** command in the [Remote menu](#) when a transfer window is the active window and no files are selected
- When you click the **Edit** button in the [transfer window toolbar](#) and no files are selected

Normally the **Edit** command operates on the selected files in the file list. But when no files are selected, you can instead create a new text file with the **Edit New Text File** command.

When you choose **Edit New Text File**, Fetch prompts you for a filename, and then opens a new document in the most appropriate text editor for that kind of file. The editor that opens depends on the extension of the filename you entered; different kinds of text files may have different editors. After you use the editor's **Save** command to save your changes, Fetch automatically uploads the new text file to the server.

Note that only text files can be created this way. Even if the filename you enter has an extension for another kind of file, Fetch will open a text editor instead.

See the [Edit](#) command help topic for more information about editing and other variations of the **Edit** command.

The **Edit Other** command is a special version of the [Edit](#) command that prompts you for the name of an existing file to edit.

The **Edit Other** command is available:

- When you hold down the Shift key and choose **Edit Other** from the [Remote menu](#)
- When you hold down the Shift key and click the **Edit** button in the [transfer window toolbar](#)

Normally the **Edit** command operates on the selected files in the file list. But when you hold down the Shift key while choosing **Edit**, the command becomes **Edit Other** and you can enter the name or path of an existing file on the server to edit.

When you choose **Edit Other**, Fetch prompts you for the name of a file on the server, and then opens that file in the editor for that kind of file (as specified in the [Transfer Options](#) pane of the [info window](#)). After you use the editor's **Save** command to save your changes, Fetch automatically uploads the changed file back to its original location on the server.

The file must already exist on the server; you cannot create new files with the **Edit Other** command. You can create new text files on a server by using the [Edit New Text File](#) command.

If the editor for the kind of file whose name you've entered is not available, or if Fetch cannot determine what kind of file it is, Fetch will ask you which editor you want to use.

See the [Edit](#) command help topic for more information about editing and other variations of the **Edit** command.

The **Edit Shortcut** command lets you modify an existing [shortcut](#).

This command is available as:

- The **Edit Shortcut** command in the [Shortcuts menu](#)
- The **Edit Shortcut** command in the [shortcut list window contextual menu](#)

The **Edit Shortcut** command displays the [Edit Shortcut dialog](#), which contains text fields describing where and to what the shortcut refers. You can edit the information and change the shortcut without having to create an entire new one from scratch.

The **Edit Shortcut** command is only enabled when a shortcut is selected in the [Fetch Shortcuts](#) window or another [shortcut list window](#).

To delete a shortcut, select the shortcut in the Fetch Shortcuts window or another shortcut list window, and choose **Edit > Delete** .

Related Topics:

- [Shortcuts](#)
- [Edit Shortcut](#)
- [Save Shortcut](#)
- [Save Droplet Shortcut](#)
- [Fetch Shortcuts window](#)
- [Shortcut list windows](#)

The **Edit With** command is a special version of the [Edit](#) command that lets you specify a different editor for making changes to the selected file.

This command is available:

- When you hold down the Option key and choose **Edit With** from the [Remote menu](#)
- When you hold down the Option key and click the **Edit** button in the [transfer window toolbar](#)
- When you hold down the Option key and choose **Edit With** from the [file list contextual menu](#)
- When you hold down the Option key and choose **Edit With** from the [shortcut list window contextual menu](#)

Normally the **Edit** command edits files using the editor specified in the [Transfer Options](#) pane of the [info window](#).

However, if you hold down the Option key while choosing **Edit**, the command becomes **Edit With**, and you can choose a different editor to use. This is useful if Fetch does not use the application you would like to use for editing a certain kind of file, or if you would like to use a different editor temporarily.

Select a file in a file list, hold down the Option key and then choose **Edit With** menu item or click the **Edit** button in the toolbar. Fetch will ask which application you would like to edit the selected files with, and you can check the **Remember this editor choice** checkbox if you want Fetch to remember this choice for the future. Then Fetch will download the file and open it in the chosen editor.

You will also be asked to choose which editor to use for a file if Fetch could not determine what kind of file you asked it to edit, or if this kind of file does not have an editor specified for it.

You can also choose a different editor to use temporarily with the [Edit With submenu](#) in the **Remote** menu or the contextual menus.

See the [Edit](#) command help topic for more information about editing and other variations of the **Edit** command.



## [Fetch Help](#) > [Commands](#) > **Edit With** submenu

The **Edit With** submenu lets you temporarily use a different editor for making changes to the selected file.

This command is available:

- The **Edit With** submenu in the [Remote menu](#)
- The **Edit With** submenu in the [file list contextual menu](#)
- The **Edit With** submenu in the [shortcut list window contextual menu](#)

Normally the **Edit** command edits files using the editor specified in the [Transfer Options](#) pane of the [info window](#). The default editor for a kind of file is usually the same as the application that would normally open that kind of file, although in certain cases the editor may be different from the normal opening application because that application cannot edit files.

The **Edit With** menu is useful if you would like to use a different editor temporarily.

Select a file in a file list, and then choose an application from the **Edit With** menu. Fetch will download the file and open it in the chosen editor.

This editor choice is temporary. The next time you edit that kind of file, the default editor for that kind of file will be used. (The current default editor is listed at the top of the **Edit With** menu.) You can permanently change the editor for a certain kind of file in the [Transfer Options](#) pane of the [info window](#), or by using the [Edit With](#) command and checking the **Remember this editor choice** checkbox.

See the [Edit](#) command help topic for more information about editing and other variations of the **Edit** command.

The **Enable Encryption / Disable Encryption** commands turn encryption on or off during an FTP session that is using FTP with TLS/SSL or Kerberos (GSSAPI or KClient) security.

This command is available as:

- The **Enable Encryption / Disable Encryption** commands in the [Remote menu](#)

When you use [FTP with TLS/SSL](#) or [Kerberos](#) (GSSAPI or KClient) to connect to a server, by default all data you exchange between your Macintosh and the server is encrypted. This means the information exchanged is scrambled so that it can only be unscrambled with a secret key, which is only known by your Macintosh and the server. This scrambling process takes some time, and may slow down sessions depending on the speed of your Macintosh and the load on the server.

For that reason, Fetch makes it possible to turn encryption off when using FTP with TLS/SSL or Kerberos (GSSAPI or KClient), in case speed is a higher priority than privacy. For example, you might disable encryption for a large download that does not need to be protected.

Note that passwords are always exchanged securely in order to keep them from being captured on the network, regardless of whether encryption is on or off.

Some FTP with TLS/SSL servers do not support encrypting transfers; you will get an error if you try to enable encryption when communicating with them.

[SFTP](#) connections also encrypt the data, but you cannot disable encryption for SFTP.

You can choose the initial state of encryption when connecting using the [New Connection dialog](#) or when [creating a shortcut](#) by checking the **Enable Encryption** checkbox in the dialog.

Related topics:

- [Security](#)
- [Kerberos](#)
- [FTP with TLS/SSL](#)
- [SFTP](#)
- [New Connection dialog](#)
- [Security Preferences](#)



The **Enter Serial Number** command lets you enter your serial number in order to license Fetch.

This command is available as:

- The **Enter Serial Number** command in the [Fetch menu](#)

When you choose the **Enter Serial Number** command, a dialog appears with fields to enter your serial number name and serial number. You must enter both pieces of information to license Fetch.

You received these numbers in your license email or letter, or if you bought Fetch from the webstore, they were displayed when your purchase was completed. Your serial number should start with the letters "FETCH" and consist of 30 letters and numbers. Some methods of purchasing Fetch automatically enter your serial number for you.

See the [Licensing Fetch](#) help topic for information about purchasing and licensing Fetch.

The **Enter Serial Number** command is disabled if you've already entered a serial number. If for some reason you need to enter a serial number again, you can hold down the Option key when selecting the **Fetch** menu to allow you to choose the menu command again.

The **Find** command searches for text in most Fetch windows.

This command is available as:

- The **Find** command in the [Edit menu](#)
- The **Find Again** command in the [Edit menu](#)

When you choose the **Find** command, the Find window appears. Enter the text you want to search for in the Find window's edit field.

When a transfer window is active, **Find** searches through the displayed item names for the desired text. When a text window (including the [Fetch Transcript](#) window) is active, **Find** searches the displayed text. When a shortcut list window (including the [Fetch Shortcuts](#) window) is active, **Find** searches the names and other properties of the shortcuts.

Two checkboxes in the Find window can change how it searches for text:

- **Ignore case** – When this box is checked, searches are not case sensitive; for example, searching for "dog" will match "dog," "Dog," or "DOG." When this box is unchecked, only exact matches for capitalization are found. Checked by default.
- **Match entire words** – When this box is checked, searches will only match words that are not part of other words; for example, "dog" will not match "doghouse." When this box is unchecked, text searches will match parts of words; for example, "dog" will match "dog," "doghouse," or "boondoggle." Unchecked by default.

If the first match isn't what you were looking for, use the **Find Again** command to find the next match. You can also return to the Find window and click the **Find** button again to find the next match.

The Find command only searches the names of items currently displayed in the file list; it does not search the contents of folders in the file list. If you're looking for a file and aren't sure where on the server it is, the [Search](#) command may be able to search all files on a server and help you find it.

Related topics:

- [Search](#)

The **Format** submenu lets you choose the format Fetch will use when putting or uploading files.

This command is available as:

- The **Format** submenu in the [Remote menu](#)
- The **Format** button pop-up menu in the [transfer window toolbar](#), if you've added it

When you put or upload a Macintosh file to a server, the best format to use depends on whether the file contains special Macintosh information, and whether the file is a text or binary file. Fetch's default upload format is **Automatic**, which examines each file and chooses the best format for it. **Automatic** works well almost all of the time, however, in some cases you may need to explicitly choose another format if the **Automatic** setting is not doing what you want or expect. See the [upload formats](#) help topic for a detailed discussion of which format to use when.

When you choose an upload format from the **Format** menu, it applies only to the active [transfer window](#) until you close it. Different transfer windows may have different format settings. The initial upload format for new transfer windows is specified by the **Default upload format** preference in the [Upload Preferences](#) pane.

The [Put](#) files and folders prompt also has a **Format** menu. The format you choose in the **Put** prompt only applies for that one upload, and does not change the setting of any transfer windows.

Upload formats are related to, but different from [download modes](#). Download modes tell Fetch how to transfer files from a server (as text or binary), they do not tell Fetch anything about what format the files are in. Upload formats tell Fetch what format to use when storing files on a server and how to transfer them to the server.

Related topics:

- [Upload formats](#)
- [Stuffit and Archive formats](#)
- [Download modes](#)

The **Get** command downloads single or multiple files or folders from a server to your Macintosh.

This command is available:

- As the **Get** or **Get Other** command in the [Remote menu](#)
- As the **Get** button in the [transfer window toolbar](#)
- As the **Get** command in the [file list contextual menu](#)
- As the **Get** command in the [shortcut list window contextual menu](#)
- By dragging files from a Fetch file list to a Macintosh folder or disk

Typically you select the files or folders you want to download in the file list before choosing the command. Multiple items can be selected by holding down the Shift or Command keys while clicking the names.

If you hold down the Shift key while choosing this command, it changes to **Get Other**, and Fetch will prompt you for the name of a file or folder to download. This can be useful if the item you want to download is in a different folder.

You can also type or paste a Fetch address (URL) into the **Get** prompt. This lets you download a file or folder from any server, regardless of whether you are currently connected to that server. See the [Uniform Resource Locators](#) topic for more information on URLs.

You can also get files by dragging the selected items to a folder or disk on the Macintosh desktop (or to the desktop itself).

The names of the downloaded files will be set automatically, based on the names that the files have on the server. All format conversions will also be performed automatically, as specified by the [download mode](#) and the decoding settings in the [Transfer Options](#) pane.

The files will be downloaded to the folder specified by the **Save downloaded files to** preference in the [Download Preferences](#) pane (by default, the download folder is set to either the Downloads folder or the Desktop). If **Save downloaded files to** is set to **Ask every time**, you will be prompted to choose a folder on your Macintosh in which to save the files and folders.

You can temporarily override the **Save downloaded files to** preference by holding down the Option key while choosing **Get**. The command will change to [Get As](#), and Fetch will prompt you to choose a folder on your Macintosh to which it should download the selected files and folders.

If a file with the same name as the file you are downloading exists in the folder you're downloading it to, Fetch will give the downloaded file a new name and leave the existing file alone, unless you turn on the **Overwrite files with conflicting names** preference in the Download Preferences pane, in which case the downloaded file will replace the existing one.

The progress of the file transfer, including the name of the file being transferred, is shown in the [status pane](#) of the transfer window. You can stop the transfer by clicking the **Stop**  button.

You can set files to [open automatically after downloading](#) using the [Transfer Options](#) pane of the [info window](#). You can also [change the application downloaded files will open with](#) in the Transfer Options pane.

Additional preferences that affect downloading of files and folders can be found in the [Download Preferences](#) pane.

Fetch tries to detect stalls and network or server errors and automatically resume downloads for you, but you can also [resume the download](#) of a file manually if necessary.

Related commands and topics:

- [Download modes](#)
- [Download Preferences](#)
- [Transfer Options](#)
- [Get As](#)
- [Put](#)



The **Get As** command is a special version of the [Get](#) command that prompts you to choose a folder on your Macintosh, and then downloads single or multiple files or folders from a server to that folder.

This command is available:

- When you hold down the Option key and choose **Get As** from the [Remote menu](#)
- When you hold down the Option key and click the **Get** button in the [transfer window toolbar](#)
- When you hold down the Option key and choose **Get As** from the [file list contextual menu](#)
- When you hold down the Option key and choose **Get As** from the [shortcut list window contextual menu](#)

Normally the **Get** command downloads files to the folder specified by the **Save downloaded files to** preference in the [Download Preferences](#) pane. The **Get As** command lets you temporarily override that preference by allowing you choose a different folder.

You must select the files or folders you want to download in the file list before choosing the **Get As** command. Multiple items can be selected by holding down the Shift or Command keys while clicking the names.

If you hold down the Shift key in addition to the Option key while choosing this command, it changes to **Get Other As**, and Fetch will prompt you for both the name of a file or folder to download, and the folder on your Macintosh to download it to. This can be useful if the item you want to download is in a different folder.

If you want to be prompted to choose a folder on your Macintosh in which to save downloaded files and folders every time you use the **Get** command, set the **Save downloaded files to** preference in the [Download Preferences](#) pane to **Ask every time**. Then you do not need to hold down the Option key every time.

You can also choose the folder to which files are downloaded by dragging the selected items to any folder or disk on the Macintosh desktop (or to the desktop itself).

For more information, see the [Get](#) command help topic.

The **Get Info** command displays an [info window](#), which shows detailed information about remote files or folders.

This command is available as:

- The **Get Info** command in the [Remote menu](#)
- The **Get Info** button in the [transfer window toolbar](#)
- The **Get Info** command in the [file list contextual menu](#)

To display an info window, select an item (or items) in a file list and choose **Remote > Get Info**.

The info window displays information about the name, kind, size, modification date, location, ownership and permissions, and transfer options of the selected item or items. You can rename items, set permissions, and change the transfer options using the info window.

You can use **Get Info** to find out how much space your files are taking up on your server, or how close you are to your server storage limit. When you select a set of files and folders, choose **Get Info**, and click the **Calculate** button, Fetch calculates the disk space taken up not only by those items, but by the items inside the folders too. Either select all the files and folders in a remote folder, or select the parent folder of the items in question, choose **Get Info** and click **Calculate**. The combined size of all items will be displayed in the info window. For many items, or folders with a large number of items inside them, this may take a while.

See the [info window](#) topic for more information.

Related topics:

- [Permissions](#)
- [Transfer Options](#)
- [Renaming files and folders](#)

 [Fetch Help](#) > [Commands](#) > **Go to Folder**

The **Go to Folder** command lets you change to another folder on the server by typing a path (either full or relative), rather than navigating through the hierarchy one folder at a time.

This command is available as:

- The **Go to Folder** command in the [Go menu](#)
- The **Go to Folder** button in the [transfer window toolbar](#), if you've added it

See the [changing folders](#) help topic for other ways to move between folders.

The **Home** command takes you to your home folder on a server — the folder that you would be taken to when you do not specify an initial folder or path when connecting.

This command is available as:

- The **Home** command in the [Go menu](#)
- The **Home** button in the [transfer window toolbar](#), if you've added it

When you do not specify an initial folder or path when making a new connection in Fetch, a server will take you to the home folder for the username you used to connect to that server. For instance, on a web server, the home folder is often the folder where your webpages are stored.

The **Home** command provides a quick way to return your home folder after you have navigated to other folders.

A home folder may not be the same as the root folder on the server (the very top folder on the server that contains all the other folders), although the home folder may be the highest-level folder you can access because you may not have permission to any enclosing folders of the home folder.

See the [changing folders](#) help topic for other ways to move between folders.



## [Fetch Help](#) > [Commands](#) > **Install Dashboard Widget**

The **Install Dashboard Widget** command installs a fresh, updated copy of the [Fetch widget](#).

This command is available as:

- The **Install Dashboard Widget** command in the [Fetch menu](#)

The Fetch widget provides easy drag-and-drop uploading of items to a specific server and displays the progress of the transfer.

Normally you are asked if you want to install the Fetch widget the first time you open Fetch. If you have already installed the Fetch widget, the widget is updated to the latest version automatically when you open Fetch.

Use the **Install Dashboard Widget** command if you declined installation of the Fetch widget when you first ran Fetch, or to install a new copy of the Fetch widget if the widget seems to be broken or misbehaving.

When you choose **Install Dashboard Widget**, if a copy of the Fetch widget is not already open, Fetch will display Dashboard and a new copy of the Fetch widget. Otherwise, Fetch will just display a dialog confirming that the installation was successful.

For more information, see the [Fetch widget](#) topic.

## [Fetch Help](#) > [Commands](#) > List of shortcuts

The list of shortcuts at the bottom of the **Shortcuts** menu provides quick access to the [shortcuts](#) listed in the [Fetch Shortcuts](#) window as well as [Bonjour](#) servers. Choose one of the shortcuts from the menu to open it.

This command is available as:

- The list of shortcuts at the bottom of the [Shortcuts menu](#)
- The **Shortcuts** submenu in the Fetch Dock icon's menu

You can also open shortcuts in the [New Connection dialog](#) using the Shortcuts  pop-up menu.

To edit the list of shortcuts, choose **Shortcuts** > **Show Shortcuts** and then use the **Shortcuts** > **New Shortcut**, **Shortcuts** > **Edit Shortcut**, and **Edit** > **Delete** commands. You can also create shortcuts by dragging items from a file list or pasting Fetch addresses (URLs) into the Fetch Shortcuts window.

Bonjour servers, if any are available, are identified by the Bonjour icon  before their name. When choosing a Bonjour server from the list of shortcuts, Fetch automatically tries to log you in to the Bonjour server with your current username. To use a different username, hold down the Option key when choosing the server from the menu, which will display the New Connection dialog with the information for connecting to the Bonjour server. Edit the username, and then connect. See the [Bonjour](#) topic for more information about Bonjour servers.

Holding down the Option key while choosing any shortcut will display the New Connection dialog filled out with the information from that shortcut.

Related topics:

- [Creating a shortcut](#)
- [Deleting or changing a shortcut](#)
- [Fetch Shortcuts window](#)
- [New Shortcut](#)
- [Edit Shortcut](#)
- [New Shortcut List](#)

The **Make Shortcut** command lets you create a new [shortcut](#) from the New Connection dialog. A shortcut is an easy way to access a remote file, folder, or server quickly without typing in connection information each time.

This command is available as:

- The **Make Shortcut** command in the Shortcuts  pop-up menu of the [New Connection dialog](#)

When you choose **Make Shortcut** from the Shortcuts pop-up menu of the New Connection dialog, Fetch displays a dialog asking you for a name for the shortcut and whether to make this the default shortcut. The name can be anything that makes it easy for you to identify the shortcut later on. When you check the **Make this the default shortcut** checkbox, the New Connection dialog will always start filled out with the information from this shortcut.

The shortcut is not created until you click the **Connect** button in the New Connection dialog. You can enter the information either before choosing **Make Shortcut**, or afterwards.

After you choose **Make Shortcut**, a checkmark will appear next to the menu item, and the menu item name will change to **Make Shortcut: My Shortcut name** or **Make Default Shortcut: My shortcut name**, depending on whether you checked the **Make this the default shortcut** checkbox. The Shortcuts pop-up menu icon will change to a heart with a plus  to indicate that a shortcut is going to be created.

If you change your mind and do not want to create a shortcut after all, choose the **Make Shortcut** menu item again. It will be unchecked, revert to saying simply **Make Shortcut**, and no shortcut will be created.

If you click **Cancel** in the New Connection dialog instead of connecting, the shortcut will not be created.

After the shortcut is created, it will appear in the Shortcuts pop-up menu, the [Fetch Shortcuts](#) window, and the **Shortcuts** menu.

If you want to change the default shortcut later on, use the **Default shortcut** pop-up menu in the [General Preferences](#) pane.

See the [creating a shortcut](#) help topic for other ways to create shortcuts, and the [deleting or changing a shortcut](#) help topic for how to remove or edit them after you've created them.

Related Topics:

- [Shortcuts](#)
- [New Shortcut](#)
- [Edit Shortcut](#)
- [Fetch Shortcuts window](#)
- [Making Fetch start with information for your server or website](#)



The **Mirror** command lets you copy only the new and changed files from a Macintosh folder to a server folder, or vice versa. You can also use it to copy new and changed files between two server folders.

This command is available as:

- The **Mirror** command in the [Remote menu](#)
- The **Mirror** button in the [transfer window toolbar](#), if you've added it

You could use the [Put](#) command to upload all the files that make up a website, but if only a few files have changed since the last upload, it isn't necessary or efficient to upload the entire site again. Instead, you can use the **Mirror** command to upload only new and changed files.

When you choose the **Mirror** command, Fetch displays the [Mirror window](#), where you specify the source and destination folders to mirror between. They can be one folder on your computer and one folder on a server, or two server folders.

You can also use the **Mirror** command to completely synchronize two folders, deleting files in the destination folder that do not exist in the source folder in addition to copying new and changed files. You should use this feature with caution; see the [Mirror window](#) help topic for more information.

You can save the settings from the Mirror window as a [mirror document](#), and simply open that document to restore the settings or start mirroring automatically.

For more information about mirroring and the Mirror window, see the [introductory](#) or [in-depth](#) Mirror window help topics, or the [mirroring a local folder to a remote folder](#) topic.

Related topics:

- [Get](#)
- [Put](#)

The **Mode** submenu lets you choose the mode (automatic, text, or binary) Fetch will use when getting or downloading files.

This command is available as:

- The **Mode** submenu in the [Remote menu](#)
- The **Mode** button pop-up menu in the [transfer window toolbar](#), if you've added it

The **Mode** menu lets you specify whether Fetch will download files as text files, binary files, or decide automatically. Most of the time the default setting of **Automatic** will do the right thing. However, if you know what kind of file you are getting, and know that it is likely to be misinterpreted by the **Automatic** mode setting, you can choose the appropriate mode instead. See the [download modes](#) help topic for a more detailed discussion of which mode to use when.

When you choose a setting from the **Mode** menu, it applies only to the active [transfer window](#) until you close it. Different transfer windows may have different mode settings. The initial mode setting for new transfer windows is specified by the **Default download mode** preference in the [Download Preferences](#) pane.

Download modes are related but different from [upload formats](#). Download modes tell Fetch how to transfer files from a server (as text or binary), they do not tell Fetch anything about what format the files are in. Upload formats tell Fetch what format to use when storing files on a server and how to transfer them to a server.

Related topics:

- [Download modes](#)
- [Upload formats](#)

 [Fetch Help](#) > [Commands](#) > **New Connection**

The **New Connection** command displays the New Connection dialog, where you can enter the information necessary to connect to a file transfer server.

This command is available as:

- The **New Connection** command in the [File menu](#)

The New Connection dialog is displayed automatically when you start Fetch, unless you have unchecked the **Show New Connection dialog at startup** preference in the [General Preferences](#) pane.

For more information, see the [New Connection dialog](#) help topic.

You can also open new connections by:

- choosing a shortcut from the list of shortcuts in the [Shortcuts](#) submenu
- choosing a server from the **File** > [Recent Connections](#) submenu
- double-clicking a shortcut in the [Fetch Shortcuts](#) window or another [shortcut list window](#)
- opening a [shortcut document](#) or [droplet shortcut](#)

The **New Folder** command creates a new folder on the server.

This command is available as:

- The **New Folder** command in the [Remote menu](#)
- The **New Folder** button in the [transfer window toolbar](#)
- The **New Folder** command in the [file list contextual menu](#)

In each case, you will be prompted for a name to give the folder. Unless you specify otherwise (by using a path), the folder will be a subfolder of the current folder.

You must have the appropriate permissions to create a folder on the server, and the server must support folders, to use this command. Otherwise, you will receive an error message.



The **New Shortcut** command lets you create a new [shortcut](#) — a quick way to access a remote file, folder, or server without typing in connection information each time.

This command is available as:

- The **New Shortcut** command in the [Shortcuts menu](#)
- The **New Shortcut** button in the [transfer window toolbar](#), if you've added it
- The **New Shortcut** command in the [file list contextual menu](#)
- The **New Shortcut** command in the [shortcut list window contextual menu](#)

The **New Shortcut** command displays the [New Shortcut dialog](#), which contains text fields that you must fill out to create the shortcut. Depending on how you brought up the dialog, the fields may already be filled out for you; for instance, if you've selected a file in a file list and choose **New Shortcut**, information to create a shortcut to that file will be automatically put in the dialog.

If a single file or folder is selected when you choose the **New Shortcut** command, the information in the New Shortcut dialog will create a shortcut that refers to that item. If multiple files or folders are selected when you choose the command, the information in the New Shortcut dialog will refer to the folder containing those items instead.

Newly created shortcuts are always listed in the [Fetch Shortcuts](#) window, unless a different [shortcut list window](#) is the active window.

See the [New Shortcut dialog](#) help topic for more information about the dialog.

You can also create new shortcuts by dragging an item from a [file list](#) to the Fetch Shortcuts window or another shortcut list window, or with the **Make Shortcut** command in the [New Connection dialog](#). See the [creating a shortcut](#) help topic for more information.

See the [deleting or changing a shortcut](#) help topic for information on how to remove or edit shortcuts after you've created them.

Related Topics:

- [Shortcuts](#)
- [Edit Shortcut](#)
- [Save Shortcut](#)
- [Save Droplet Shortcut](#)
- [Fetch Shortcuts window](#)
- [Shortcut list windows](#)

 [Fetch Help](#) > [Commands](#) > **New Shortcut List**

The **New Shortcut List** command opens a new [shortcut list window](#), where you can create a list of shortcuts separate from the main [Fetch Shortcuts](#) window.

This command is available as:

- The **New Shortcut List** command in the [Shortcuts menu](#)

Shortcut list windows can be saved in a document for opening later.

Shortcut list windows other than the Fetch Shortcuts window are not reopened automatically.

See the [shortcut list window](#) topic for more information.

Related commands:

- [New Shortcut](#)
- [Edit Shortcut](#)
- [Save Shortcut](#)
- [Save Droplet Shortcut](#)

 [Fetch Help](#) > [Commands](#) > **Open**

The **Open** command prompts you to open one or more previously saved [shortcut documents](#), [shortcut list documents](#), or [mirror documents](#).

This command is available as:

- The **Open** command in the [File menu](#)

**File > Open** only opens saved Fetch documents from your Macintosh.

To open a file shown in a transfer window, use the [Quick Look](#) command ([View](#) on Mac OS X 10.4) or [Edit](#) command in the [Remote menu](#); or download the file by selecting it and clicking on the [Get](#) button, and then opening it in the appropriate application. (You can also set certain types of files to automatically open after downloading using the [Transfer Options](#) pane of the [info window](#).)

To open a shortcut in the [Fetch Shortcuts](#) window or a [shortcut list window](#), double-click the shortcut, or select the shortcut and press Command–Down arrow. You can also choose the shortcut from the list of shortcuts in the **Shortcuts** menu.

Related commands:

- [Quick Look](#) (or [View](#) on Mac OS X 10.4)
- [Edit](#)
- [Open Folder](#)



The **Open Folder** command displays a remote folder's contents in a file list.

This command is available:

- As the **Open Folder** command in the [Go menu](#)
- As the **Open Folder** command in the [file list contextual menu](#)
- As the **Open Folder** command in the [shortcut list window contextual menu](#)
- By swiping down on laptop trackpads that support gestures

The **Open Folder** command makes the selected folder the current folder and displays its contents in the file list.

You can also open a folder by double-clicking on it, typing Command-O, or typing Command-Down arrow.

To open a folder in a new transfer window, hold down the Command key while double-clicking on the folder.

The **Open Folder** command is only available when a single folder is selected.

In the [shortcut list window contextual menu](#), the **Open Folder** command opens the folder referenced by the selected shortcut in a transfer window, opening a connection to the server containing the folder if necessary.

To open a file shown in a transfer window, use the [Quick Look](#) command ([View](#) on Mac OS X 10.4) or [Edit](#) command in the [Remote menu](#); or download the file by selecting it and clicking on the [Get](#) button, and then opening it in the appropriate application. (You can also set certain types of files to automatically open after downloading using the [Transfer Options](#) pane of the [info window](#).)

Related commands:

- [Quick Look](#) (or [View](#) on Mac OS X 10.4)
- [Edit](#)
- [Open](#)

 [Fetch Help](#) > [Commands](#) > **Page Setup**

The **Page Setup** command shows the standard Page Setup dialog so you can set page attributes such as paper size, orientation, and scaling for printed documents.

This command is available as:

- The **Page Setup** command in the [File menu](#)

For more help with **Page Setup**, see the Mac OS X help topic on the [Page Setup dialog](#).

Related commands:

- [Print](#)



The **Parent Folder** command takes you to the parent folder (enclosing folder) of the folder you're currently viewing, and updates the file list.

This command is available:

- As the **Parent Folder** command in the [Go menu](#)
- As the **Parent** button in the [transfer window toolbar](#), if you've added it
- By swiping up on laptop trackpads that support gestures

If you are at the root folder, the **Parent Folder** command will be disabled.

If you do not have permission to view the parent folder, you will receive an error when you choose the **Parent Folder** command.

You can drag files and folders to the **Parent** button in the toolbar to move them up one level to the parent folder of the current folder without having to open a new window or connection.

See the [changing folders](#) help topic for other ways to move between folders.

The **Paste** command lets you paste text; paste Fetch addresses (URLs) into the New Connection dialog; paste shortcuts that you previously cut or copied into shortcut list windows; or paste files for uploading or copying to a new location.

This command is available as:

- The **Paste** command in the [Edit menu](#)
- The **Paste** command in the [file list contextual menu](#)
- The **Paste** command in the [shortcuts list window contextual menu](#)

You can use the **Paste** command to paste text in any place where you can edit text.

You can paste Fetch addresses (FTP URLs) you previously copied anywhere Fetch addresses are accepted, such as the [New Connection dialog](#), the [Get](#) prompt, or the [Go to Folder](#) prompt.

You can also paste a Fetch address (URL) into a [shortcut list window](#) or into the [New Shortcut dialog](#) to create a shortcut.

If you've set up [WebView](#) for a server, you can also paste web addresses (HTTP URLs) for that server in the New Connection dialog, the New Shortcut dialog, or shortcut list windows.

In [shortcut list windows](#), you can paste shortcuts you previously cut or copied.

You can use the **Copy** and **Paste** commands to upload files and folders in Fetch, or to copy files and folders from one folder to another on the same server, or from one server to another. See the [uploading using copy and paste](#) or [copying files using copy and paste](#) help topics for more information.

Related topics:

- [Cut](#)
- [Copy](#)
- [Duplicate](#)
- [URLs](#)



The **Path** button lets you go to any folder that contains the current folder.

This command is available:

- As the **Path** button in the [transfer window toolbar](#)
- By Command-clicking the title of the transfer window

When you click the **Path** button, a pop-up menu of all the folders containing the current one appears. Choose a folder to go to that folder.

If you are at the root folder of the server, the **Path** button will only contain the root folder because there are no enclosing folders.

If you do not have permission to view the one of the folders in the path, you will receive an error if you choose it from the menu.

The name of the server is displayed at the bottom of the pop-up menu. You cannot choose it; it is there as a reminder.

See the [changing folders](#) help topic for other ways to move between folders.

 [Fetch Help](#) > [Commands](#) > **Print**

The **Print** command shows the standard Print dialog and lets you print the contents of the Fetch Transcript window, any [text window](#), or any [media window](#) (Mac OS X 10.4 only).

This command is available as:

- The **Print** command in the [File menu](#)

You cannot print any other windows in Fetch.

Related commands:

- [Page Setup](#)



## [Fetch Help](#) > [Commands](#) > **Purchase**

The **Purchase** command lets you buy a license for Fetch, and automatically enters your serial number after you've finished the purchase process.

This command is available as:

- The **Purchase** command in the [Fetch menu](#)

When you choose the **Purchase** command, a window guiding you through the purchase process appears. When you finish the purchase process, your serial number will be automatically recorded in this copy of Fetch, and you will also receive an emailed receipt — please keep the receipt in case you ever need to reenter your serial number.

For more information about Fetch pricing, see the [Licensing Fetch](#) help topic.

The **Purchase** command does not appear in the menu if you've already entered a serial number. If you need to access the **Purchase** menu item again, holding down the Option and Shift keys while selecting the **Fetch** menu will allow you to choose it again. You can also visit the [Fetch Softworks website](#) if you wish to purchase additional copies.

## [Fetch Help](#) > [Commands](#) > **Purchase Upgrade**

The **Purchase Upgrade** command lets you buy an upgrade for Fetch, and automatically enters your new serial number after you've finished the purchase process.

This command is available as:

- The **Purchase Upgrade** command in the [Fetch menu](#)

When you choose the **Purchase Upgrade** command, a window appears to guide you through the upgrade purchase process. When you finish the purchase process, your new serial number will be automatically recorded in this copy of Fetch, and you will also receive an emailed receipt — please keep the receipt in case you ever need to reenter your serial number.

For more information about Fetch pricing, see the [Licensing Fetch](#) help topic.

The **Purchase Upgrade** command only appears when you have a licensed older version of Fetch, and upgrading to the new version requires a purchase. The **Purchase Upgrade** command does not appear in the menu if you've already entered a serial number valid for this version of Fetch, but you can access the [Purchase](#) menu item again by holding down the Option and Shift keys while selecting the **Fetch** menu. Or you can visit the [Fetch Softworks website](#) if you wish to purchase additional copies of Fetch.

Related topics:

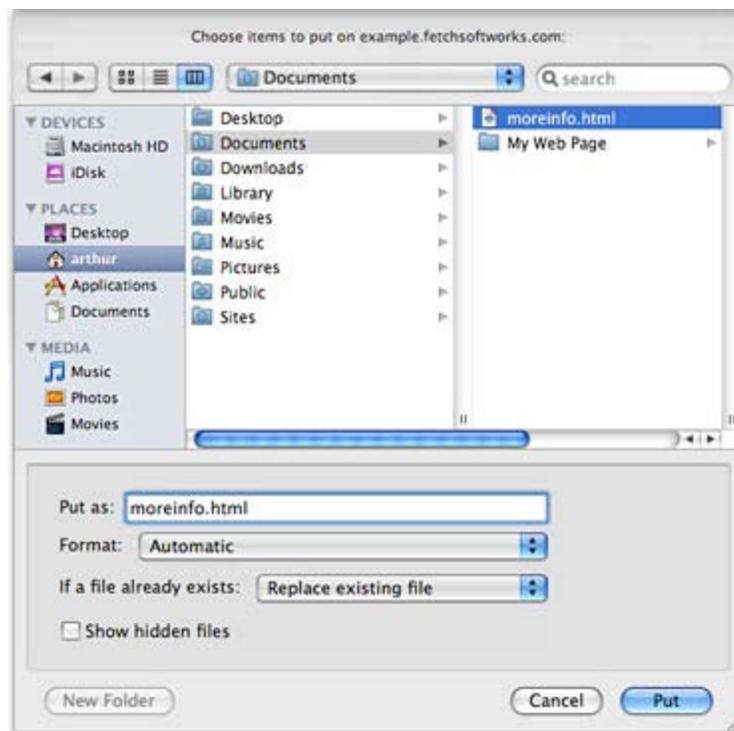
- [Upgrading from an earlier version of Fetch](#)

The **Put** command uploads one or more files or folders from your Macintosh to a server.

This command is available:

- As the **Put** command in the [Remote menu](#)
- As the **Put** button in the [transfer window toolbar](#)
- As the **Put** command in the [shortcut list window contextual menu](#)
- By dragging files from a Macintosh folder or disk to a Fetch file list

When you use the **Put** menu command or toolbar button, you will be prompted to select the files and folders you want to upload. You will also be prompted to choose an [upload format](#) to use, and what to do if files with the same names already exist on the server.



You can select either a single file or folder to upload, or multiple files and folders. To select multiple files and folders, hold down the Shift key while clicking to select adjacent files, or hold down the Command key to select non-adjacent files.

The names of uploaded files will be set automatically, based on the names the files have on your Macintosh. Extensions may be added to the names of the uploaded files, as specified by the **Add filename extensions when uploading** preferences in the [Upload Preferences](#) pane. If a file of the same name already exists on the server, it will be overwritten without warning, unless you choose a different option from the **If a file already exists** pop-up menu in the Put dialog.

If you select a single file in the Put dialog, you can change the name it will be given on the server with the **Put as** field. Be sure to pick a name that will be acceptable to the server (some servers require filenames that are shorter than Macintosh filenames, and most do not accept the use of certain characters).

If you're uploading a folder, Fetch will create a new folder on the server and put the contents of the Macintosh folder in that remote folder, unless you've chosen one of the [archive formats](#). If the Macintosh folder has subfolders, matching subfolders will be created on the server as well. If the server already has folders with the same names as the Macintosh folders, the files you're uploading will be added to the existing remote folders. The entire contents of existing remote folders will not be replaced — only files inside the folders that have the same names as files being uploaded will be replaced.

You can also upload files and folders by dragging them from the Finder to a Fetch file list, or Fetch folder shortcut. In that case, you will not be prompted for any information; Fetch will use the [upload format](#) specified in the **Format** submenu in the [Remote menu](#), and if there's a name conflict, Fetch will replace the existing files with the new files.

You can also [upload using Copy and Paste](#), or create a [droplet shortcut](#) for uploading.

Uploading folders will not work with servers that do not support subfolders (such as VM/CMS).

If you've chosen one of the archive formats as your upload format, all the files you're uploading will be encoded into a single file and that single file will be put on the server, instead of folders or multiple files being put on the server. You can specify a name for the new archive in the **Put as** field.

The progress of the file transfer, including the name of the file being transferred, is shown in the [status pane](#). You can halt the transfer by clicking the **Stop**  button.

The formats you can use when putting a file are [Automatic](#), [Text](#), [MacBinary](#), [Binary \(Raw Data\)](#), [BinHex](#), [AppleSingle](#), [Gzip](#), [Stuffit X Archive](#), [Gzipped Tar Archive](#), and [Zip Archive](#). The Automatic upload format is usually the best choice — it chooses the best format for the file based on the file's extension and other properties. The initial setting of the Put dialog's **Format** menu is the same as the transfer window you're uploading to. You can choose the default upload format for new transfer windows in the [Upload Preferences](#) pane. You can also change a transfer window's upload format with the **Format** submenu in the [Remote menu](#). See the [upload formats](#) help topic or a format's individual help topic for more information.

In the Put dialog, choices for what to do if a file already exists on the server are:

- **Replace existing file** – Existing files with the same name as files being uploaded will be overwritten without warning.
- **Pick a different name** – Only available with SFTP connections. If a file you're uploading has the same name as an existing remote file, Fetch will pick a new name for the new file.
- **Resume upload of file** – If an existing remote file and a new file have the same name, Fetch will assume the file was only partially transferred, and add the end of the new file to the end of the existing file on the server, so that they are the same length. This can be useful if your upload was interrupted for some reason. You can only resume uploads of binary files that are not encoded; if you select some other sort of file, this option will be disabled. For more information, see the [resuming uploads](#) topic.

Check the **Show hidden files** to show Macintosh files that are not normally displayed in the Finder, such as those that begin with a period or that are marked as invisible.

Preferences that affect uploading of files and folders can be found in the [Upload Preferences](#) pane.

Related topics:

- [Upload Preferences](#)
- [Upload formats](#)
- [Automatic upload format](#)
- [Stuffit and Archive formats](#)
- [Resuming uploads](#)
- [Get](#)
- [Edit](#)

The **Quick Look** command lets you display files on a server without leaving Fetch. Many different kind of files can be displayed, including text files, images, sounds, movies, PDFs, and Microsoft Word and Excel files.

The **Quick Look** command is only available on Mac OS X 10.5 and later. If you are on Mac OS X 10.4, see the [View](#) command help topic instead.

This command is available:

- As the **Quick Look** command in the [Remote menu](#)
- As the **Quick Look** button in the [transfer window toolbar](#)
- When files are selected and you press the Space bar
- As the **Quick Look** command in the [file list contextual menu](#)
- As the **Quick Look** command in the [shortcut list window contextual menu](#)

Select a file in a file list, then choose **Quick Look**. Fetch will download the file and display it in a Quick Look window.

To close a Quick Look window, press the Space bar again, choose the **Quick Look** command again, or click in the window's close box.

Fetch's Quick Look command can display any file that the Finder's Quick Look command can display. (Note that not all features of Quick Look may be available when using Quick Look in Fetch on Mac OS X 10.5.)

If you double-click on a file in a Quick Look window, that file will open in the application that normally opens that kind of file. Note that if you make changes to the file after doing this, they will not be saved back to the server. If you want to make changes to files and have them saved back to the server, use the [Edit](#) command instead.

Quick Look windows will float above all other Fetch windows until closed. You cannot open more than one Quick Look at a time. If you select a different file while the Quick Look window is open, Fetch will download that file and display it in the Quick Look window instead.

If you want to view a file as plain text — either because Quick Look is displaying the file as formatted text, or because Quick Look does not recognize the file as a text file — use the [View as Text](#) command instead.

Related commands:

- [Edit](#)
- [View as Text](#) (on Mac OS X 10.5 and later)
- [View](#) (on Mac OS X 10.4)

 [Fetch Help](#) > [Commands](#) > **Quit**

The **Quit** command closes your current connections and quits Fetch.

This command is available as:

- The **Quit Fetch** command in the [Fetch menu](#)

When you quit Fetch, any open connections will be closed. If you have file transfers in progress, Fetch will warn you and give you a chance to cancel your request to quit. If you have any unsaved shortcut list windows, Fetch will ask you if you want to save the changes before quitting.

Tip: If you happen to accidentally quit Fetch in the middle of a transfer, you may be able to resume the transfer, rather than starting from the beginning of the file. For more information about this potential time-saver, see the [resuming an interrupted download](#) or [resuming uploads](#) help topics.



## [Fetch Help](#) > [Commands](#) > **Recent Connections submenu**

The **Recent Connections** submenu provides quick access to servers and shortcuts that you've recently visited. Choose one of the listed servers or shortcuts to open a connection to it.

This command is available as:

- The **Recent Connections** submenu in the [File menu](#)
- The **Recent Connections** submenu in the Fetch Dock icon's menu

You can also open your recent connections in the [New Connection dialog](#) using the Recent Connections  pop-up menu.

The Recent Connections items are in the form of either "shortcut name—folder" or "username@hostname—folder" if there is no shortcut for that hostname. When a shortcut name is displayed, or you connected as a guest to a server, the username will be omitted. If you last visited your home folder on the server, the folder name will be omitted. If you have connected to the same server via two different [connection types](#), the connection type will be listed in parentheses after the hostname.

When you choose an item from this menu, you will be taken to the folder you most recently visited on the server, as indicated by the folder name listed in the menu item.

If you hold down the Option key while choosing a recent connection item from this menu, instead of the connection being opened directly, the New Connection dialog will be displayed, filled out with the information for that recent connection item.

The last 10 connections you've made will be remembered. Choose **Clear Menu** from the submenu to empty the list of recent connections.

Related commands:

- [New Connection](#)



## [Fetch Help](#) > [Commands](#) > Recently visited folders

Fetch remembers the remote folders you've recently visited on each server so you can return to them quickly.

Recently visited folders are available:

- In the [Go menu](#)
- When you click the **Recent** button in the [transfer window toolbar](#)

Choosing one of the remote folder names from either location makes that folder the current folder and updates the file list.

The number of folders in the recently visited folders list depends on how many folders you've visited, and there may be no folders listed if you haven't made any connections or if you've used the **Clear Menu** command at the bottom of the **Go** menu. Fetch remembers a list of folders for each server you've used recently.

You can clear the list of folders with the **Clear Menu** command at the bottom of the **Go** menu. Otherwise, Fetch remembers the folders you've visited on each server in the last two weeks.

See the [changing folders](#) help topic for other ways to move between folders.

Related commands:

- [Clear Menu \(Go Menu\)](#)

 [Fetch Help](#) > [Commands](#) > Refresh

The **Refresh** command updates the file list with the latest information from the server.

This command is available as:

- The **Refresh** command in the [View menu](#)
- The **Refresh**  button in the [transfer window](#), on the right in the info bar just above the file list
- The **Refresh** command in the [file list contextual menu](#)

Fetch caches and reuses file lists for speed, which can sometimes cause it to miss new changes to the list. If you just uploaded a file and it does not appear in the file list, use the **Refresh** command to fix the problem. See the [file list](#) help topic for more information about file list caching.



The **Resume Download** command lets you resume a partially completed download.

This command is available as:

- The **Resume Download** command in the [Remote menu](#)
- The **Resume Download** button in the [transfer window toolbar](#), if you've added it

If a download is interrupted unexpectedly or accidentally, it is sometimes possible for Fetch to resume it without re-downloading the part of the file that you already have.

Since Fetch tries to detect stalls and network or server errors and automatically resume downloads for you, you should not need this feature under most circumstances.

See the [resuming an interrupted download](#) help topic and the [Resume Download window](#) help topic for more information.

Related topics:

- [Get](#)



## [Fetch Help](#) > [Commands](#) > **Save**

When a [shortcut list window](#) other than the Fetch Shortcuts window is active, **Save** lets you save the changes to it, or save it to a [shortcut list document](#) if it hasn't been previously saved.

This command is available as:

- The **Save** command in the [File menu](#), when a shortcut list window other than the Fetch Shortcuts window is active

To save a remote file or remote folder to your Macintosh, use the [Get](#) button in the transfer window toolbar, or the [Get](#) menu command.

**Save** is not available in the [Fetch Shortcuts](#) window because changes to it are saved automatically. To save the contents of the Fetch Shortcuts window to a new shortcut list document, use the **Save a Copy As** command.

To save the contents of a [text window](#), [media window](#) (Mac OS X 10.4 only), or the [Fetch transcript](#) window, use the [Save a Copy As](#) command.

To save the contents of a [Quick Look](#) window (Mac OS X 10.5 and later only), double-click the contents of the window to open them in an external application, and use that application's **Save** command.

Related Commands:

- [Save a Copy As](#)
- [Save Shortcut](#)
- [Save Droplet Shortcut](#)
- [Save As](#)
- [Get](#)

 [Fetch Help](#) > [Commands](#) > **Save As**

When a [shortcut list window](#) is active, **Save As** lets you save it to a new [shortcut list document](#).

This command is available as:

- The **Save As** command in the [File menu](#), when a shortcut list window is active

Even if you have previously saved a shortcut list window to a shortcut list document, the **Save As** command lets you save it to a new, different shortcut list document.

**Save As** is not available in the [Fetch Shortcuts](#) window; use **Save a Copy As** instead.

To save the contents of a [text window](#) or the [Fetch transcript](#) window to a new local file, use the [Save a Copy As](#) command.

Related Commands:

- [Save](#)
- [Save a Copy As](#)
- [Save Shortcut](#)
- [Save Droplet Shortcut](#)

The **Save Droplet Shortcut** command creates a [droplet shortcut](#). You can drag files to a droplet shortcut in the Finder or Dock to upload those files, or double-click it to open a remote folder in Fetch.

This command is available as:

- The **Save Droplet Shortcut** command in the [File menu](#)

The **Save Droplet Shortcut** command is only available when a [transfer window](#) is active and no files or a single folder is selected, or when the [Fetch Shortcuts](#) window or another shortcut list window is active and a single folder shortcut is selected. Otherwise, when a file or multiple items are selected in a transfer window, the [Save Shortcut](#) command will be available instead.

When a folder is selected in the [file list](#), **Save Droplet Shortcut** creates a droplet shortcut that will upload files dragged to it, or open the selected folder. If no items are selected in the file list, it creates a droplet shortcut that will upload files or open the folder currently being viewed in the transfer window.

When a shortcut is selected in the Fetch Shortcuts window, **Save Droplet Shortcut** creates a droplet shortcut equivalent to that shortcut — the droplet shortcut will upload files or open the folder referred to by that shortcut.

Once the droplet shortcut is created, just drag files and folders to its icon, and Fetch will open and automatically start uploading the files to the folder represented by the droplet shortcut. If you double-click the droplet shortcut instead, it will open the folder it represents in Fetch, but will not upload anything.

You can choose an [upload format](#) for the droplet shortcut to use by selecting a format from the **Upload format** menu in the Save Droplet Shortcut dialog. All items dropped on this particular droplet shortcut will always be uploaded in that format. Usually you want **Automatic**, the default choice.

You can also choose to have Fetch quit after an upload started by the droplet shortcut completes by checking the **Quit Fetch after uploading** checkbox in the Save Droplet Shortcut dialog.

For more information, see the [droplet shortcuts](#) help topic.

To create a shortcut in the [Fetch Shortcuts](#) window or another [shortcut list window](#) instead, drag a file or folder from a file list to a shortcut list window, or use the [New Shortcut](#) command.

Related Topics:

- [Creating a shortcut](#)
- [Save Shortcut](#)
- [Shortcut documents](#)
- [Shortcuts](#)
- [Save](#)

The **Save Shortcut** command creates a shortcut document, a file that you can double-click in the Finder to download a file in Fetch.

This command is available as:

- The **Save Shortcut** command in the [File menu](#)

The **Save Shortcut** command is only available when a [transfer window](#) is active, and a file or multiple items are selected. Otherwise, when a folder or no files are selected, the [Save Droplet Shortcut](#) command will be available instead.

When a file is selected in the [file list](#), **Save Shortcut** creates a [shortcut document](#) that refers to the selected item.

When multiple files and folders are selected, you can save a single shortcut document that refers to all of them. When you open that shortcut file, all the files will be downloaded and all the folders will be opened in Fetch. For more information, see the [shortcut document](#) help topic.

When a folder or no items are selected, you can use the [Save Droplet Shortcut](#) command to create a droplet shortcut. You can drag files to a droplet shortcut in the Finder or Dock to upload those files, or double-click it to open that folder in Fetch. For more information, see the [droplet shortcut](#) help topic.

To create a shortcut in the [Fetch Shortcuts](#) window or another [shortcut list window](#), drag a file or folder from a transfer window to a shortcut list window, or use the [New Shortcut](#) command.

To save a shortcut list window, use the [Save](#) command.

Related Topics:

- [Creating a shortcut](#)
- [New Shortcut](#)
- [Shortcut documents](#)
- [Save Droplet Shortcut](#)
- [Shortcuts](#)
- [Save](#)

## [Fetch Help](#) > [Commands](#) > **Save a Copy As**

The **Save a Copy As** command lets you save to a local disk the contents of a [text window](#), [media window](#), or the [Fetch Transcript](#) window; it also lets you save a copy of the [Fetch Shortcuts](#) window.

This command is available as:

- The **Save a Copy As** command in the [File menu](#)

Use the **Save a Copy As** command to save the contents of a remote text file or remote media file to a local disk, or to save a copy of the Fetch Shortcuts window to a new shortcut list document.

If you want to [back up your shortcuts](#), use the **Save a Copy As** command to save a copy of them to a new file.

To save a [shortcut list window](#) other than Fetch Shortcuts as a new shortcut list document, use the **Save As** command.

To save the contents of a [Quick Look](#) window (Mac OS X 10.5 and later only), double-click the contents of the window to open them in an external application, and use that application's **Save** command.

Related Topics:

- [Save](#)
- [Save As](#)
- [Save Shortcut](#)
- [Save Droplet Shortcut](#)
- [Backing up your shortcuts](#)



[Fetch Help](#) > [Commands](#) > Search (Send FTP Commands)

The Search command, part of the [Send FTP Commands](#) command, attempts to search an entire server for files and folders whose names contain a certain string, and creates a shortcut list with the results.

This command is available as:

- The **Send FTP Commands** command in the [Remote menu](#)

Some FTP servers support a command ("SITE INDEX") to search an entire FTP server for file and folder names that include a certain string. This makes it possible to find a file on an FTP server without knowing which folder it is in, and without manually navigating through all the folders.

Enter the name (or partial name) of files you are looking for into the Search field of the **Send FTP Commands** dialog, and if the Search command is supported by your server, Fetch creates a [shortcut list window](#) with shortcuts for all of the matches found. You can then open or download the matches by double-clicking the shortcuts.

Unfortunately, many servers do not support this command, in which case you will receive an error instead. Search and **Send FTP Commands** are not available when you are connected to a server using SFTP.

To find text in a Fetch window, use the [Find](#) command instead.

Related topics:

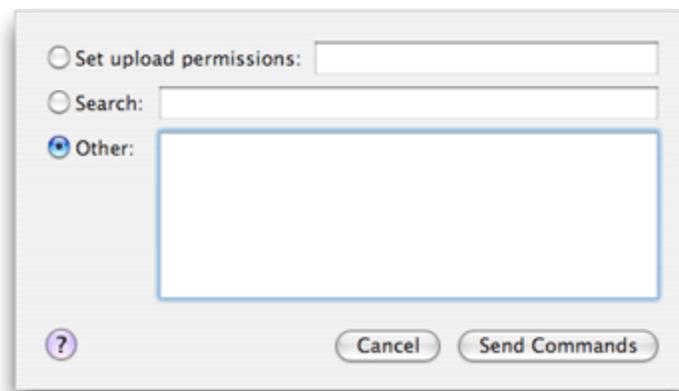
- [Find](#)
- [Send FTP Commands](#)

The **Send FTP Commands** command lets you send arbitrary FTP commands to an FTP server, with special support for the Search and Set Upload Permissions commands.

This command is available as:

- The **Send FTP Commands** command in the [Remote menu](#)
- The **Send FTP Commands** button in the [transfer window toolbar](#), if you've added it

Most file transfer tasks can be handled with Fetch's regular buttons and menu items, but some FTP servers implement non-standard commands, or require use of standard commands in non-standard situations. In these cases, you can send an arbitrary raw FTP command with the **Send FTP Commands** menu command. It prompts you for one or more FTP commands, and passes them along to the server without interpretation.



The **Send FTP Commands** command is not available when you are connected to a server using SFTP, because SFTP does not support sending raw commands.

A couple of commands, [Search](#) and [Set Upload Permissions](#), that are of special interest to select users can be accessed with the **Send FTP Commands** dialog as well. Search attempts to search a server for files and folders whose name contains a certain string. Set Upload Permissions lets you specify the permissions to be used for files uploaded to an FTP server in octal format. For more information on these fields, see their respective help topics.

Arbitrary commands should be entered into the Other field of the **Send FTP Commands** dialog. The Search field expects a string such as you might find in a filename, and the Set Upload Permissions field expects numbers.

Note that arbitrary FTP protocol commands often differ from the UNIX FTP client commands. For example, "cd" is not a valid FTP command, but "CWD" is. Likewise, "ls" and "dir" are not valid commands; the proper commands to list files are "NLST" and "LIST", respectively. FTP commands can be entered in upper or lower case.

If you send a command to list files ("NLST" or "LIST"), Fetch will display the file list in a new text window.

Examples of commands that may be useful (these are specific to UNIX servers):

LIST -lt     - lists files sorted by date  
NLST \*.hqx - lists files that end in ".hqx"  
HELP        - lists the commands supported by this server

To see the results of some FTP commands, you may need to open the [Fetch Transcript](#) window. For example, the list of commands generated by "HELP" appears in the Fetch Transcript window.

**Warning:** if you enter a command that starts a file transfer, it will almost certainly not work and you may have to close the connection.

Command-line FTP clients have a command called "quote" that is equivalent to the **Send FTP Commands** command in Fetch. So if you have been instructed to type "quote blah blah", in Fetch you should enter just "blah blah" in the Other field of the **Send FTP Commands** dialog.

Related topics:

- [Search](#)
- [Set Upload Permissions](#)

 [Fetch Help](#) > [Commands](#) > **Set Font and Size**

The **Set Font and Size** command lets you set the font and size of text in [text windows](#), [shortcut list windows](#), and transfer window [file lists](#). It also lets you set the tab width used in text windows.

This command is available as:

- The **Set Font and Size** command in the [View menu](#)

When you choose **Set Font and Size**, a dialog appears asking you what font and size you'd like the current window to use. Normally the new font and size settings affect only the current window, and only until you close it. If you want all windows of the same kind as the current window to use the new font and size settings permanently, check the **Make these settings default** checkbox.

For instance, to change the font and size used by all file lists, choose **Set Font and Size** when a transfer window is active, and check **Make these settings default**. This will not affect the permanent font setting for text windows or shortcut list windows.

In text windows, the **Set Font and Size** command also allows you to set the width of the tab stops used to display tabs in the text.

The Set Upload Permissions command, part of the [Send FTP Commands](#) command, lets you specify the permissions to be used for files uploaded to an FTP server. Permissions restrict which users can read, write, or execute files.

This command is available as:

- The **Send FTP Commands** command in the [Remote menu](#)

To set upload permissions, choose **Send FTP Commands**, enter the desired permissions in octal form (usually three digits, such as "755") in the Set Upload Permissions field, and click **OK**. Then upload the files you want to have those permissions. The timeframe these permissions stay in effect is dependent on the server. They may only affect the next upload, or they may stay in effect until the end of your session with the server.

Access to files on FTP servers is typically governed by permission settings. Some FTP servers (typically those running the UNIX operating system) make it possible for FTP clients to change those permissions, provided that the user is the owner of the files in question. Set Upload Permissions specifies the permissions to be used for files that are yet to be uploaded. Use the [Get Info](#) command to set the permissions of files and folders already on the server.

To verify that permissions are set correctly, you can use the [Get Info](#) command or the [View Verbose File List](#) command. Fetch may not display the true permissions of just uploaded items correctly until you click the [Refresh](#)  button in the transfer window (if the info window shows all unchecked checkboxes, the permissions are most likely not up to date).

Note that FTP servers won't set execute permissions via Set Upload Permissions, no matter what is entered in the dialog, so the actual permissions will be whatever is entered, minus the execute permissions.

If you receive an error when using the Set Upload Permissions command, or the permissions are not set as you specified, it's likely that the server does not support the UNIX-derived permission command necessary ("SITE UMASK"). If the server does not support setting Execute permissions on uploads, use [Get Info](#) to set the permissions after you've uploaded the files.

Set Upload Permissions and **Send FTP Commands** are not available when you are connected to a server using SFTP.

Related topics:

- [Get Info](#)
- [Send FTP Commands](#)
- [View Verbose File List](#)
- [Putting Folders and Files](#)

 [Fetch Help](#) > [Commands](#) > **Stop**

The **Stop** command halts the current transfer or action.

This command is available as:

- The **Stop** command in the [View menu](#)
- The **Stop** button in the [status pane](#) of the [transfer window](#)

The **Stop** command and **Stop**  button halt the transfer or action currently in progress. Stopping actions may leave them half-completed, for example, if you were uploading a file, an incomplete file may be left on the server; if you were downloading a set of files or folders, some but not all of the files may be on your local drive; or if you stopped a listing of files, the file list may show only a partial list of files.

You can also stop the current action by typing Command-**.** (Command-period).

If you stopped the transfer of a file, you may be able to resume it later — see the [resuming an interrupted download](#) or [resuming uploads](#) help topics.



## [Fetch Help](#) > [Commands](#) > **View**

The **View** command lets you view files from a server without leaving Fetch. Kinds of files that can be viewed include text files and images, sounds, movies, or any other sort of file that can be displayed by Apple's QuickTime software.

The **View** command is only available on Mac OS X 10.4. If you are on Mac OS X 10.5 or later, see the [Quick Look](#) command help topic instead.

This command is available as:

- The **View** command in the [Remote menu](#)
- The **View** button in the [transfer window toolbar](#)
- The **View** command in the [file list contextual menu](#)
- The **View** command in the [shortcut list window contextual menu](#)

Select a file in a file list, then choose **View**. Fetch will download the file and display it in a new [text window](#) or new [media window](#) as appropriate.

If Fetch is determining the kind of file incorrectly, that is, showing a media file in a text window, you can hold down the Option key while choosing **View**. The command changes to [View As](#) and Fetch asks you whether you want to view the file as text or as media. Note that there are some kinds of files that Fetch cannot display internally, such as Microsoft Word files. To view such a file, download the file and open it in the correct application.

If you hold down the Shift key while choosing this command, the command changes to [View Other](#) and Fetch will prompt you enter the name of a file to view. **View Other** is only available when a transfer window is the active window.

Related commands:

- [Edit](#)
- [Quick Look](#) (on Mac OS X 10.5 and later)
- [View as Text](#) (on Mac OS X 10.5 and later)

The **View As** command is a special version of the [View](#) command that lets you specify whether you want to view a file as text or as media.

The **View As** command is only available on Mac OS X 10.4. If you are on Mac OS X 10.5 or later, see the [Quick Look](#) or [View as Text](#) command help topics instead.

This command is available:

- When you hold down the Option key and choose **View As** from the [Remote menu](#)
- When you hold down the Option key and click the **View** button in the [transfer window toolbar](#)
- When you hold down the Option key and choose **View As** from the [file list contextual menu](#)
- When you hold down the Option key and choose **View As** from the [shortcut list window contextual menu](#)

Normally the **View** command automatically determines what kind a file is, but **View As** is useful if Fetch determines the kind of file incorrectly, such as showing a media file in a text window.

Select a file in a file list, hold down the Option key and then choose **View As**. Fetch will ask you whether you want to view the file as text or media, and then download the file and display it in a new [text window](#) or new [media window](#) as appropriate.

You will also be asked to choose how you want to view a file if Fetch could not determine what kind of file you asked it to view (probably because the filename has an unknown extension, or no extension at all).

Note that there are some kinds of files that Fetch cannot display internally, such as Microsoft Word files. To view such a file, download the file and open it in the correct application.

For more information, see the [View](#) command help topic.

Related commands:

- [Edit](#)
- [Quick Look](#) (on Mac OS X 10.5 and later)
- [View as Text](#) (on Mac OS X 10.5 and later)

## [Fetch Help](#) > [Commands](#) > **View as Text**

The **View as Text** command lets you view files from the server as plain text without leaving Fetch.

The **View as Text** command is only available on Mac OS X 10.5 and later. If you are on Mac OS X 10.4, see the [View](#) or [View As](#) command help topics instead.

This command is available as:

- The **View as Text** command in the [Remote menu](#)
- The **View as Text** button in the [transfer window toolbar](#), if you've added it
- The **View as Text** command in the [file list contextual menu](#)
- The **View as Text** command in the [shortcut list window contextual menu](#)

Select a file in a file list, then choose **View as Text**. Fetch will download the file and display it in a new [text window](#).

**View as Text** is an alternative to the [Quick Look](#) command. Normally, **Quick Look** displays certain kinds of text files, such as HTML or RTF files, as formatted text instead of the raw plain text. You can use **View as Text** to force Fetch to display those files as plain text instead (so that you see the raw HTML and RTF codes instead). It is also useful if you want to look at the contents of a text file with a filename extension that Quick Look does not recognize as a text file.

You can also use **View as Text** to view the contents of multiple files at once, and while Fetch is in the background. Quick Look is limited to displaying one file at a time, and the Quick Look window is only displayed when Fetch is in the foreground.

To view media files such as images, movies, or sounds, use the [Quick Look](#) command (Mac OS X 10.5 and later) or [View](#) command (Mac OS 10.4) instead.



[Fetch Help](#) > [Commands](#) > **View Other**

The **View Other** command is a special version of the [View](#) command that prompts you for the name or path of a file to view, and whether to view that file as text or as media.

The **View Other** command is only available on Mac OS X 10.4. If you are on Mac OS X 10.5 or later, see the [Quick Look](#) command help topic instead.

This command is available:

- When you hold down the Shift key and choose **View Other** from the [Remote menu](#)
- When you hold down the Shift key and click the **View** button in the [transfer window toolbar](#)

Normally the **View** command operates on the selected files in the file list, but **View Other** is useful if you know the name of a file and don't want to scroll through the file list looking for it. Fetch will select the correct "view as" button in the View Other dialog based on the extension of the filename you type in, but you can override its choice if desired.

Note that there are some kinds of files that Fetch cannot display internally, such as Microsoft Word files. To view such a file, download the file and open it in the correct application.

**View Other** is available when you hold down the Shift key and choose the **View** command or click the **View** toolbar button. It is only available when a transfer window is the active window.

For more information, see the [View](#) command help topic.

 [Fetch Help](#) > [Commands](#) > **View Verbose File List**

The **View Verbose File List** command shows an extended text-only file list for the current folder of the active transfer window in a separate [text window](#).

This command is available as:

- The **View Verbose File List** command in the [Remote menu](#)

The extended file list will often include information, such as file permissions, that Fetch does not display in its regular file list. However, it is in a less user-friendly format than the Fetch file list, and is not interactive.

The **WebView** command displays the currently selected items in your web browser or prompts you to set up WebView for the current server, if necessary.

This command is available as:

- The **WebView** command in the [Remote menu](#)
- The **WebView** button in the [transfer window toolbar](#)
- The **WebView** command in the [file list contextual menu](#)
- The **WebView** command in the [shortcut list contextual menu](#)

WebView lets you set up a correspondence between files you are viewing in Fetch and the address of the webpage they are part of, so that you can easily review changes to your site in a web browser and copy the web address of those files. For more information about WebView, see the [WebView](#) help topic.

To use WebView, select an item in a Fetch file list, and click the **WebView** button in the toolbar or choose **Remote > WebView**.

If you have already set up WebView, a page corresponding to that item will open in your web browser. If you selected an HTML file representing a webpage, you will see the webpage, or if you selected an image file, you'll see the image. If you selected a folder, what you see in the web browser depends on how your server is configured; you may see the contents of the folder, you may see a webpage that's inside that folder (if that folder has its own index file), or you may get an error, because web browsers aren't allowed to see the contents of that folder directly.

If you have not already set up WebView for this folder or one of its parent folders, the [WebView dialog](#) will appear, requesting the web address of your server. Before you can use WebView, you must tell Fetch the web address of your server. (The information for some popular servers is built into Fetch, and if you are using one of them, WebView will work automatically; but in most cases Fetch needs your input.) For more information about what to enter in the dialog, see the [WebView dialog](#) help topic.

The WebView dialog will only appear the first time you choose WebView; once WebView is set up for that server, you do not need to enter this information again. The next time you choose **WebView** when connected to the same server, Fetch will show the selected files or folder in your web browser instead of the WebView dialog.

Related topics:

- [WebView](#)
- [Setting up WebView](#)
- [WebView dialog \(web address only\)](#)
- [WebView Preferences](#)
- [Copy Web Address](#)

The following topics offer information about concepts and terms used in Fetch and its documentation. If you don't see what you're looking for, check the [Using Fetch](#) list of topics, or use the Search field above.

- [Active and Passive FTP](#)
- [AppleSingle format](#)
- [Automatic upload format](#)
- [Automator actions](#)
- [BinHex format](#)
- [Binary \(Raw Data\) format](#)
- [Bonjour](#)
- [Character encoding & translation](#)
- [Connection types](#)
- [Download modes](#)
- [Drag-and-drop](#)
- [Droplet shortcuts](#)
- [FTP](#)
- [FTP with TLS/SSL \(FTPS\)](#)
- [Gzip format](#)
- [Gzipped Tar Archive format](#)
- [Kerberos](#)
- [Keyboard shortcuts](#)
- [Keychain support](#)
- [MacBinary format](#)
- [Mirror documents](#)
- [One-Time Password](#)
- [Padlock icon](#)
- [Permissions](#)
- [S/Key](#)
- [SFTP](#)
- [Scripting](#)
- [Security](#)
- [Server-to-server transfers](#)
- [Shortcut documents](#)
- [Shortcut list documents](#)
- [Shortcuts](#)
- [Software update](#)
- [Software update privacy](#)
- [Stuffit X Archive format](#)
- [Stuffit, Zip, and archive formats](#)
- [Text upload format](#)
- [Transfer Options](#)

[Uniform Resource Locators \(URLs\)](#)

[Upload formats](#)

[WebView](#)

[Zip Archive format](#)

Whenever Fetch retrieves a file list or transfers a file, it has to set up a new connection between your Macintosh and the FTP server. There are two ways it can do this, called active FTP and passive FTP. These aren't really different types of FTP, they're different ways of opening an FTP connection between the FTP client (Fetch) and the FTP server. Sometimes you'll be told to use one or the other, especially with respect to firewalls, routers, and NAT (Network Address Translation) devices. Active FTP is how FTP originally worked; passive FTP was developed to solve problems using active FTP with firewalls and NAT devices.

By default, Fetch uses its Automatic Passive Mode feature to determine whether passive or active FTP will work best for the connection you're using, and you don't have to worry about which to choose — your connection should just work successfully. The Automatic Passive Mode feature is always enabled, you do not have to turn it on.

However, in case Fetch's auto-detection capability doesn't work with your setup, this topic describes the difference between active and passive FTP, how they relate to using FTP with firewalls and NAT devices, when you should use active or passive, and what to do if neither one works. This topic only applies when connecting using FTP; SFTP does not have a notion of active or passive connections.

## **Ports & Firewalls**

Your computer often talks to several different servers over the network simultaneously. For instance, you may be receiving email from one server while loading a webpage from another server, or you may be transferring files using FTP with several different servers.

In order to keep track of all these conversations, called connections, computers assign numbers to each connection. These are called port numbers, or ports for short.

A firewall is hardware or software that attempts to protect computers by preventing computers outside the firewall from starting connections with computers inside the firewall. Generally, your local network is inside, or behind, the firewall, whereas the rest of the Internet is outside the firewall. Firewalls only allow computers inside the firewall to start connections with outside computers. A firewall is like having a telephone system that doesn't accept any incoming calls, but that lets you place outgoing calls. As a result, you can still have conversations with other people, you've just got to be the one who starts them.

## **Active vs. Passive FTP**

To make an FTP connection, the server needs to know on which port to talk to your Macintosh.

In active FTP, which was designed before firewalls were common, Fetch tells the server "this is the port you should talk to me on," and the server attempts to connect to that port. This is like Fetch giving the server a phone number to call your computer at. The firewall blocks incoming calls, so you get an error when trying to open a connection because Fetch never hears from the server.

In passive FTP, Fetch asks the server to pick a port, and then connects to the server on that port. This is like Fetch asking at what phone number it can call the server. Since Fetch makes the call, the firewall allows it, and you are all set to transfer files.

NATs (Network Address Translation) are devices that allows multiple computers to share a single IP address. NAT devices can have issues similar to firewalls with connections coming in

from the outside. In fact, some NAT devices also act as firewalls, so passive FTP also works better with NAT devices.

### When To Use Passive FTP

Fetch's Automatic Passive Mode feature tries to automatically determine whether it should use passive or active FTP. You do not need to do anything to enable this feature. If the auto-detection does not seem to be working, you should use passive FTP at all times unless you receive an error trying to make a connection and have eliminated all other possible problems (see the [Troubleshooting FAQ](#)).

If you are trying to connect to a non-standard FTP port while behind a NAT device, the NAT device may not translate the request properly if you're using active FTP. You should use passive FTP in this case (or switch to connecting with SFTP).

FTP connections using [Kerberos](#) from behind a NAT device require that you use passive FTP. Even then, the server may require special changes to support Kerberos from behind a NAT device. Consult your server maintainer if necessary.

### When To Use Active FTP

You should use active FTP in the following cases:

- The FTP server you are trying to connect to does not support passive FTP connections.
- The FTP server you are trying to connect to is behind a firewall, router, or NAT device itself.

You can tell Fetch to use active FTP for connection by unchecking the **Use passive mode transfers** preference in the [General Preferences](#) pane.

### What To Do If Neither Active or Passive FTP Works

In some cases, you may not be able to establish a connection with either active or passive FTP. For example, this may happen if both you and the FTP server are behind firewalls or NAT devices. In this case, you should try the following:

- Try connecting to the server using SFTP. SFTP works quite a bit differently from FTP and does not have the same problems with firewalls and NAT devices.
- If your Mac OS X firewall is turned on, try disabling it temporarily. On Mac OS X 10.5 or later, you can [configure the firewall to allow an exception for Fetch](#).
- Contact the server maintainer and suggest they support passive FTP, or find out why passive FTP might not be working.
- [Contact Fetch support](#).

## [Fetch Help](#) > [Concepts](#) > **AppleSingle Format**

AppleSingle is Apple's official format for representing a Macintosh file on other types of computers. Fetch can get files in AppleSingle version 1 and 2 formats, and put files in AppleSingle version 2 format.

Fetch automatically recognizes and decodes AppleSingle files when downloading, although automatic recognition can be turned on and off by either:

- Choosing **Fetch > Preferences**, clicking the [Obscure](#) tab, and checking or unchecking the **Recognize and decode AppleSingle** — this will enable or disable decoding of AppleSingle files but will not affect the decoding of other formats, such as Zip or Stuffit archives;
- Or choosing **Fetch > Preferences**, clicking the [Download](#) tab, and checking or unchecking the **Allow automatic decoding of downloaded files** checkbox — this will enable or disable decoding of all file types.

While the **Automatic** download mode will usually recognize and decode AppleSingle files, if you are having problems it may be necessary to choose the **Binary** download mode temporarily.

See the [upload formats](#) help topic for information about encoding files in AppleSingle when uploading.

Related topics:

- [Upload formats](#)
- [Download modes](#)
- [Obscure Preferences](#)



The **Automatic [upload format](#)**, like the **Automatic [download mode](#)**, tells Fetch to use the information available to it to make the best choice among a number of formats when uploading a Macintosh file to a server. The formats are: Text, Binary (Raw Data), MacBinary, BinHex, AppleSingle, or Zip Archive. (Automatic upload format never uses Stuffit X Archive, Gzipped Tar Archive, or Gzip when uploading a file.)

The goals of the **Automatic** upload format are to preserve any significant special Macintosh information in a file (preserving both data and resource forks if necessary), and after that, to properly transfer the file as a text or binary file.

When the **Automatic** upload format is selected, Fetch will choose the best format for each file by following these rules in order:

- If Fetch is connected to a server that can interpret MacBinary files, Fetch transfers the file in MacBinary III format. That way, all Macintosh-specific information will be preserved.
- If Fetch is connected to a server that can interpret AppleSingle files, Fetch transfers the file in AppleSingle format. That way, all Macintosh-specific information will be preserved.
- Fetch determines if the file is a text file or a binary file. It uses information such as filename extension and the file's Finder type to make this determination.
- If a file is determined to be a text file, Fetch transfers the file in Text format, discarding any resource fork the file may have. Line ending conversion will take place on FTP connections (but not SFTP), and an extension may be appended to the file's name, depending on the **Add filename extension when uploading Text files** preference.
- If the file is determined to be a binary file, Fetch checks to see if the file has a resource fork.
- If the file has a resource fork, Fetch checks to see if the resources are significant. Resources not regarded as significant mostly consist of Finder custom icons and resources added to JPEGs and GIFs by Photoshop.
- If the file has no significant resources, or no resource fork at all, Fetch transfers the file in Binary (Raw Data) format. An extension may be appended to the file's name, depending on the **Add filename extension when uploading Other files** preference.
- If file has a resource fork with significant resources, Fetch transfers the file in Zip Archive format, so that the resources are preserved.

If you want to preserve the resources of a file that Fetch would normally discard when using Automatic upload format (such as Photoshop resources), you will have to temporarily choose another format, such as MacBinary or Zip Archive, from the **Remote > Format** menu, and then upload that file.

See the [upload formats](#) help topic for more information about each format.

Related topics:

- [Upload formats](#)
- [Upload Preferences](#)
- [Download modes](#)

Fetch includes a number of actions for Mac OS X's Automator, allowing you to create simple workflows that can automate a variety of Fetch tasks.

This help topic describes all the Fetch Automator actions in detail. If you are not familiar with Automator, you may want to start by reading the [introduction to using Automator with Fetch](#) help topic, which has a brief overview of Automator and examples of creating Automator workflows that use Fetch.

On Mac OS X 10.5 and later, the Fetch Automator actions can be found in the Internet category.

The available Fetch Automator actions are:

**Delete Fetch Items** – This action deletes the Fetch Items passed in as input. You can have the action ask for confirmation before deleting the items.

*Input:* Fetch Items. *Result:* nothing.

**Download Fetch Items** – This action downloads the Fetch Items passed in as input to the folder you specify on your Macintosh. Items are always downloaded using the Automatic [download mode](#).

*Input:* Fetch Items. *Result:* Files/Folders.

**Filter Fetch Items** – This action passes only the Fetch Items that meet the specified criterion on to the next action. You can filter by name, name extension, item type (file, folder, or alias), size, modification date, and permissions. The Name and Name Extension filters are case insensitive.

*Input:* Fetch Items. *Result:* Fetch Items.

**Generate Fetch Item Report** – This action generates a text report listing information about the Fetch Items passed in as input. This action does not create a text file on its own; to see the report, you must pass the results to another action that accepts text input, such as the TextEdit "New TextEdit Document" or "New Text File" actions.

*Input:* Fetch Items. *Result:* Text.

**Get Fetch Folder Contents** – This action returns the Fetch Items passed in as input plus the Fetch Items inside any folders in the input items. If the "Repeat for each subfolder found" box is checked, then the contents of all nested folders will be included in the results as well.

*Input:* Fetch Items. *Result:* Fetch Items.

**Get Selected Fetch Items** – This action gets the selected Fetch Items from the frontmost Fetch [transfer window](#) and passes them to the next action. It requires a Fetch transfer window with one or more items selected.

*Input:* The selected items. *Result:* Fetch Items.

**Get Specified Fetch Items** – This action passes the Fetch Items specified in the list to the next action. You can add items to the list by clicking on the Add (+) button, which displays a dialog similar to the [New Connection dialog](#). You can either enter the information necessary to

specify an item, or you can choose one of your recent connections or shortcuts from the pop-up menus. If a password is required to access the items, the password should be stored in your [keychain](#) in order to allow the workflow to operate without further input. Otherwise, Fetch will prompt you for the password and the workflow will be on hold until you enter the password.

*Input:* None, or Fetch Items. *Result:* Fetch Items.

**Move Fetch Items** – This action moves the Fetch Items passed in as input to a different remote folder, and then passes the items (in their new location) to the next action. You can choose to move the items to the items' parent folders; your home folder; the root folder of the server; a sub-folder of the items' current folder; or to a specific folder in your home folder, the items' parent folder, or the root folder.

*Input:* Fetch Items. *Result:* Fetch Items.

**Rename Fetch Items** – This action renames the Fetch Items passed in as input. You can rename the files by adding a date and time to the existing names, adding text before or after the existing names, changing the case of the letters in the names, numbering the files sequentially, or finding and replacing specific text in the file names. Each renaming method offers a variety of options for how the renaming should be done. At the bottom of the action, you will see an example of what the new names will look like after you perform this action.

If you want to change the extensions of a batch of files, you can use the "Replace Text" pane of Rename Fetch Items to accomplish this.

*Input:* Fetch Items. *Result:* Fetch Items.

**Set Fetch Item Permissions** – This action sets [permissions](#) on the Fetch Items passed in as input, and then passes the items to the next action. You can specify different permissions for files and folders, or just set permissions for one or the other. To specify the permissions, you can either enter three-digit UNIX equivalent strings directly in the text fields, or you can choose the permissions from the pop-up menus to the right of the text fields. This action will ignore alias items and leave their permissions unchanged.

*Input:* Fetch Items. *Result:* Fetch Items.

**Upload Files** – This action uploads the files and folders passed in as input to the specified Fetch folder and returns the newly created Fetch Items as the result. You can either enter the connection information for the destination, or you can choose one of your recent connections or shortcuts from the pop-up menus. (If Fetch is not already running, there may be a delay when you click the recent connections or shortcuts menus while Fetch is opened.) If a password is required to open the connection, the password should be stored in your keychain in order to allow the workflow to operate without further input. Otherwise, Fetch will prompt you for the password and the workflow will be on hold until you enter the password. In addition to specifying the connection information for the destination, you can choose the [upload format](#) that should be used when uploading the items. After the upload is complete, the action outputs a list of the files (but not the folders) that were uploaded.

*Input:* Files/Folders. *Result:* Fetch Items (files only).

In addition, you can also take advantage of Fetch's extensive [AppleScript support](#) in Automator by using the Automator "Run AppleScript" action.

You can save Automator workflows as applications, allowing you to easily create droplets to perform Fetch actions. For instance, create a workflow that consists of an Upload Files action

configured for your favorite upload location, and save the workflow as an application. Then you can just drag files and folders to the droplet, and they'll be uploaded to your server automatically without requiring further input. If you want to automatically change the files' permissions or rename the files after uploading them, add a Set Fetch Item Permissions or a Rename Fetch Items action after the Upload Files action.

Most of the Fetch actions support the "Show action when run" option, which displays an action's options as a workflow runs, so that you can change the options from their default settings when you run the workflow.

For more examples of things you could do with Automator and Fetch, see the [introduction to using Automator with Fetch](#) and [more examples of using Automator With Fetch](#) help topics.

Related topics:

- [Scripting](#)



BinHex (sometimes called BinHex4) is a format for representing a Macintosh file in text form. The Macintosh file, which may contain special Macintosh information and raw binary data, is converted to a single file consisting of series of lines, each made up of letters, numbers, and punctuation. Because BinHex files are simply text, with no special information, they can be sent through most electronic mail systems and stored on most computers. However, the conversion to text makes the file larger, so it takes longer to transmit a file in BinHex format than if the file was represented some other way. The filename extension ".hqx" usually indicates a BinHex format file.

Fetch can get and put files in BinHex format. See the [upload formats](#) help topic for information about encoding files in BinHex when uploading.

Fetch automatically recognizes and decodes BinHex files when downloading, although automatic recognition can be turned on and off by selecting a remote file whose name ends with ".hqx", choosing **Remote > Get Info** to show the [info window](#), and, in the [Transfer Options](#) pane, checking or unchecking the **Automatically decode files like this** checkbox.

Fetch automatically recognizes and decodes BinHex files when downloading, although automatic recognition can be turned on and off by either:

- Selecting a remote file whose name ends with ".hqx", choosing **Remote > Get Info** to show the [info window](#), and, in the [Transfer Options](#) pane, checking or unchecking the **Automatically decode files like this** checkbox — this will enable or disable decoding of BinHex files but will not affect the decoding of other formats, such as Zip or Stuffit archives;
- Or choosing **Fetch > Preferences**, clicking the [Download](#) tab, and checking or unchecking the **Allow automatic decoding of downloaded files** checkbox — this will enable or disable decoding of all file types.

Related topics:

- [Upload formats](#)
- [Transfer Options](#)

 [Fetch Help](#) > [Concepts](#) > **Binary (Raw Data)**

Binary (Raw Data) is one of the formats available for putting files on another computer. If this format is chosen, the data part of the Macintosh file will be uploaded without any interpretation or conversion.

Some Macintosh programs create files with all their information in the data part (such as Stuffit, Microsoft Excel, Microsoft Word). Similarly, most cross-platform standard file formats (such as GIF and JPEG image formats, and virtually every sort of file used on the World Wide Web) use only the data part. In these cases, using the Binary (Raw Data) format is a simple way to transfer files that does not result in information getting lost.

However, some Macintosh files include a resource part, and that information will be lost if you use Binary (Raw Data) format. (The file type and icon may not be preserved, among other things.) Therefore the Binary (Raw Data) format should only be used if the file is strictly a data file that will be used on the World Wide Web or can be used by some other computer (such the Windows version of Excel on a Windows PC). If the file will be retrieved only by a Macintosh, either [MacBinary](#) or [BinHex](#) format should be chosen instead.

In most cases, the [Automatic upload format](#) will choose the correct format for your files and you do not have to worry about choosing a specific one.

Related topics:

- [Upload formats](#)

Bonjour (called Rendezvous on older version of Mac OS X) is a feature of Mac OS X that lets computers automatically discover and advertise services on a network with minimal or no configuration. Bonjour is based on an official Internet standard known as Zeroconf.

Fetch will automatically find Bonjour FTP and SFTP servers on your networks. They appear in the [New Connection dialog's](#) Shortcuts  pop-up menu and in the list of shortcuts at the bottom of the **Shortcuts** menu. They are identified by the Bonjour icon  before the name.

The advantage of Bonjour is that you do not need to know a server's hostname or address to find it — servers simply appear automatically in the Fetch shortcut menus. Choosing one fills out the correct information in the New Connection dialog for Fetch to access it. Otherwise, working with Bonjour FTP and SFTP servers in Fetch is just like working with any other server — you can transfer files to and from the server, you can save a shortcut to one, etc.

When you choose a Bonjour server from the **Shortcuts** menu, Fetch automatically tries to log you in to the server with your current Mac OS X username. To use a different username, hold down the Option key when choosing the server from the **Shortcuts** menu. This will display the [New Connection dialog](#) filled out with the information for connecting to the Bonjour server. Edit the username, and then click **Connect**.

You can create Bonjour FTP and SFTP servers on a Macintosh using the **Sharing** pane of **System Preferences**. To create a Bonjour FTP server, enable FTP Access; to create a Bonjour SFTP server, enable Remote Login (SSH). Note that enabling either of these options turns that Macintosh into a full-fledged file server that can be accessed from anywhere on the Internet, as well as via Bonjour, by anyone who has an account on that Macintosh and knows that Macintosh's hostname or IP address.

Many computers, including the Macintosh, can display special characters that are important for communicating in Roman languages besides English (some examples: é ö   ), as well as characters for non-Roman languages, such as Japanese or Hebrew. Unfortunately, different computers use different codes to refer to these characters. So, unless special translation is performed, a filename that looks fine on one computer may be a garbled mess on another.

The **Preferred encoding** preference in the [Miscellaneous preferences](#) pane addresses this problem. This preference tells Fetch which character set it should assume servers are using, so that it can display filenames and file contents correctly on your Macintosh. If servers you regularly connect to show the filenames as gibberish or not in the characters you expect, you should try a different encoding.

For example, if you connect to a server that has filenames in Japanese, and your preferred encoding is **Western (ISO Latin 1)**, the filenames will be displayed in Roman characters because the Western (ISO Latin 1) encoding tells Fetch to interpret text as Roman characters instead of Japanese. This won't be too useful if you're trying to read them in Japanese. If you change your preferred encoding to **Japanese (ISO-2022-JP)**, you'll see the Japanese characters. However, the Japanese encoding is probably not the best for viewing a server with filenames all in Roman characters. This is an extreme example; for certain servers a different Western encoding may be better for displaying accented characters than the default Western encoding.

The **Preferred encoding** preference affects the display of item names in transfer window file lists, the names of files after they've been downloaded to your Macintosh (or names of files you upload to the server), and how Fetch displays text files when you use the [View](#) command on Mac OS X 10.4 (but not when you use the **Quick Look** command on 10.5 or later). However, it does not affect the contents of files at all.

If a file list contains characters that cannot be displayed using your preferred encoding, you'll receive an error and Fetch will stop trying to display the file list and disconnect from the server. To solve this problem, change your preferred encoding.

When you use the [Edit](#) command to edit a file with BBEdit 9.2 or later (or TextWrangler 3.0 or later), Fetch will pass along encoding specified by the **Preferred encoding** preference as the encoding the file should be displayed in. However, if your file already specifies its encoding with a BOM or an HTML meta tag, BBEdit will detect that and use the encoding specified by the file regardless of Fetch's **Preferred encoding** setting. Again, the contents of the file are not changed; if the encoding sent to BBEdit is not the correct one, you can use BBEdit's **Reopen Using Encoding** command (in the **File** menu) to display it in the correct encoding. Currently, no other text editors support receiving the preferred encoding setting from Fetch, although other editors may be able to detect the encoding specified by a file on their own.

**Preferred encoding** is an overall preference and cannot be specified for individual servers. If you have a particular server that requires a different encoding from your usual setting, you can change the preferred encoding temporarily to work with that server, and then set it back once you are done with that connection.

**Western (ISO Latin 1)** is the default preferred encoding because it is the most common and compatible character set. While filenames may appear garbled, they will at least be displayed. (ISO Latin 1 is the ISO 8859-1 character set.)

Other encodings of note are:

**Unicode (UTF-8)** – Use this encoding when connecting to Mac OS X, Java, or newer Windows servers.

**Western (Mac OS Roman)** – Use this encoding when connecting to servers based on Mac OS 9 or earlier.

**Western (Windows Latin 1)** – Use this encoding for connecting to older Windows servers.

Fetch may do additional translation of filenames depending on the setting of the **Encode and decode file and folder names** preference. When that preference is checked, in addition to translating to/from the preferred encoding, Fetch will encode and decode item names according to the AppleSingle 7-Bit ASCII standard: slash (/), percent (%), null, and all 8-bit extended ASCII characters will be encoded as a three-character sequence of the form %xx, where xx is the character's hexadecimal value. This is useful for preserving special characters in names on servers that don't support any sort of special characters.

Fetch supports a variety of different types of connections to servers: **FTP** (no security), **SFTP**, **FTP with TLS/SSL** (FTPS), **FTP with GSSAPI** (Kerberos), and **FTP with KClient** (Kerberos). Not all servers support all these choices; be sure the server you're connecting to supports your choice, or you will receive an error.

You can specify the connection type with the **Connect Using** pop-up menu in the [New Connection dialog](#) and [New Shortcut dialog](#). The security of a connection is indicated by a padlock icon just above the [file list](#) in [transfer windows](#) and in [shortcut list windows](#).

## FTP

The most common type of connection. A plain FTP connection, with no extra security beyond possibly asking for a password — your data and password are not protected or encrypted in any way. (Some servers use [One-time Password](#) or [S/Key](#) to protect the password.)

## SFTP

SFTP stands for "SSH File Transfer Protocol." When you connect to a server using SFTP, it uses SSH encryption to protect the connection between your Macintosh and the server. This protects your password and your data, preventing an eavesdropper from watching and learning them as they travel over the network. See the [SFTP](#) help topic for more information.

## FTP with TLS/SSL

Also known as FTPS. When you connect to a server using FTP with TLS/SSL, it uses TLS (Transport Layer Security) or SSH (Secure Sockets Layer) encryption to protect the connection between your Macintosh and the server. This protects your password and optionally your data, preventing an eavesdropper from watching and learning them as they travel over the network. Normally Fetch uses the AUTH TLS (also known as FTPES or Explicit SSL) method of establishing FTP with TLS/SSL connections; to use SSL connect (also known as Implicit SSL) select this option and enter 990 in the Port field. See the [FTP with TLS/SSL](#) help topic for more information.

## FTP with GSSAPI FTP with KClient

Connect to the server via FTP, using Kerberos for authentication. GSSAPI and KClient are interfaces to Kerberos, a network security system. When you choose one of these options, you can securely connect to a server using your Kerberos tickets instead of a password. Optionally you can also encrypt your data using Kerberos. Generally you shouldn't choose these options unless you know you're using other Kerberos services as well. See the [Kerberos](#) help topic for more information. FTP with KClient support is not available on Mac OS X 10.5 or later.

Related topics:

- [Security](#)



FTP and SFTP (the protocols Fetch uses to transfer files) divide files into two categories: text (ASCII) files and binary files. Furthermore, the Macintosh uses complex files that require the use of special formats when transferred to other types of computers. These issues must be considered in order for a file transfer to be successful. Fetch has three download modes that control the way a file is retrieved — they are titled **Automatic**, **Text**, and **Binary**.

You can set which download mode Fetch will initially use for all transfer windows with the **Default download mode** preference in the [Download Preferences](#) pane. You can change the download mode temporarily for a transfer window with the [Mode](#) submenu in the [Remote](#) menu.

If a binary file (such as a JPEG file) is downloaded as a text file, it will become corrupted and unusable, so it's important to use the right mode. Fetch has an **Automatic** download mode that works well for most situations so that you don't have to decide each time. Unfortunately, Fetch cannot always tell what sort of file it is downloading and treat it in the appropriate way, so some of the time you may have to specify what sort of file you are getting. The three possible settings are explained below.

### **Automatic**

Fetch starts out with the **Automatic** mode selected, and this mode will be the right one most of the time. When Automatic mode is selected, Fetch uses the filename extension of the file you are getting to determine what to do. For instance, text files usually have the extension ".txt," and JPEG files end in ".jpg". This information leads Fetch to download ".txt" files as text files and download ".jpg" files as binary files. Fetch comes with a default set of mappings of filename extensions to the appropriate download modes, and it also uses information from the operating system and the Transfer Options pane of the [info window](#) to decide the correct download mode. See the [Transfer Options](#) help topic for more information.

Some files do not offer any clues because they don't have extensions, and in that case Fetch will use its default fallback behavior (initially, Binary mode). You can change the fallback behavior by selecting a file without an extension, choosing **Remote > Get Info**, and in the Transfer Options pane, choosing either **Text** or **Binary** from the **Get files like this as** menu.

Fetch can automatically recognize and decode certain kinds of encoded files after downloading them — BinHex, AppleSingle, and MacBinary files, as well as Stuffit X archives, Zip archives, Gzip compressed files, and Gzipped Tar archives. When Fetch detects an encoded file, it automatically decodes the file unless you set the [Transfer Options](#) in the info window to not decode that kind of file, or if the **Allow automatic decoding of downloaded files** checkbox in the [Download Preferences](#) pane is unchecked.

The **Automatic** setting may do the wrong thing if the filename doesn't give enough information about the file kind, or suggests the wrong file kind, for example a ".txt" file that is in fact a binary file. For these situations, you must use one of the other two settings.

**Note:** One notable exception to these rules for **Automatic** mode is that if Fetch determines that the server is a Macintosh-based FTP server, it will try to transfer all files in MacBinary mode. This way, all files will appear exactly as they do on the

server.

## Text

This setting tells Fetch to assume downloaded files are text files (or BinHex files, which are a kind of text file). Text files are also known as ASCII files. When a text file is downloaded, line endings will be converted according to the **Text file line endings** preference in the [Download Preferences](#) pane. If a text file you're downloading has the sort of name usually associated with binary files (for example, ending in "bin", "Z", "sit", "gif", or "tar"), you should choose this setting to override the automatic behavior.

## Binary

This setting is for getting binary (non-text) files, specifically those files whose names do not identify them as being binary files. If you know that a file is not a text file (for instance, it is in MacBinary or AppleSingle format), choosing this setting will help ensure the proper treatment. When this mode is chosen, the downloaded file will be checked to see if it is in MacBinary or AppleSingle format.

Related topics:

- [Download Preferences](#)
- [Transfer Options](#)
- [Upload formats](#)



Fetch takes advantage of the drag-and-drop capabilities of the Mac OS to make moving files, shortcuts, text, and preference information as convenient as possible. You can use drag-and-drop in the following situations:

- To download (get) a remote file or folder, drag it from a file list or shortcut list window to a Finder icon, window, or the desktop.
- To delete a remote file or folder, drag it from a file list to Mac OS Trash icon.
- To upload (put) a file or folder, drag it to a transfer window or folder shortcut (in a shortcut list window).
- To move a file or folder into a subfolder, drag it to the subfolder icon.
- To move a file or folder up a folder level, drag it to the **Parent** button in the transfer window toolbar.
- To copy a file or folder into a subfolder, drag it to the subfolder icon with the Option key pressed.
- To copy a file or folder up a folder level, drag it to the **Parent** button in the transfer window toolbar with the Option key pressed.
- To transfer a file or folder between servers, drag it to another open transfer window.
- To create a shortcut for a remote item, drag it to a shortcut list window.
- To copy shortcuts, drag them between shortcut list windows.
- To delete a shortcut from a shortcut list window, drag it to the Mac OS Trash icon.
- To copy the URL of a remote item or shortcut, drag it to an application that accepts text drags (such as Stickies or BBEdit).
- To create a shortcut or copy a URL to the current folder of a transfer window, drag the proxy icon from a transfer window title bar to a shortcut list window to create a shortcut, to the Finder to create a droplet shortcut, or to a text editor to copy the folder's URL.
- To create a shortcut or copy a URL to the items in an info window, drag the proxy icon from an info window title bar to a shortcut list window to create a shortcut, to the Finder to create a shortcut document or droplet shortcut, or to a text editor to copy the items' URL.
- To copy text from text windows, drag the selected text to a text editor's window or to the Finder to create a clipping.
- To specify a remote folder for mirroring, drag it from a file list to a Mirror window pane.
- To specify a local folder for mirroring, drag it from the Finder to a Mirror window pane.
- To specify the download folder, drag a Finder folder icon to the **Save downloaded files to pop-up menu** in the [Download Preferences](#) pane.
- To specify the default FTP or SFTP helper application, drag an application to the **Default FTP application** or **Default SFTP application** menus in the [General Preferences](#) pane.

Droplet shortcuts are used to access remote folders from the Finder or the Dock. You can drag files to a droplet shortcut to have Fetch upload those files, or double-click it to open a remote folder in Fetch without needing to enter any connection information.

While using a droplet shortcut is similar to using [shortcuts](#) in Fetch, a droplet shortcut may save you more steps or may suit your workflow better. For instance, if there's a server that you regularly upload files to, you could create a droplet shortcut that refers to a folder on that server. Then you can just drag files and folders to the droplet shortcut in the Finder, enter your password if necessary, and Fetch will connect to the server and start uploading your files.

You can also quickly connect to the remote folder to make other changes by double-clicking the droplet shortcut. Fetch will prompt you for a password if necessary, and display the folder's contents in a Fetch window. No navigating is needed in either case.

You create droplet shortcuts with the [Save Droplet Shortcut](#) command.

To create a droplet shortcut:

1. Open a connection to the desired server.
2. In the transfer window file list, navigate to the folder you want to upload files to.
3. Do not select anything in the file list if you want to upload files to the folder currently being displayed in the transfer window; otherwise, select a folder you want to upload files to.
4. Choose **File > Save Droplet Shortcut**.
5. Optionally choose an upload format to use (usually you want the default choice of Automatic) and whether you want Fetch to quit after an upload completes.
6. Enter a name for and choose a location to save the droplet shortcut.
7. Click the **Save** button.

Once the droplet shortcut is created, just drag files and folders to its icon, and Fetch will open and automatically start uploading the files to the folder represented by the droplet shortcut. If you double-click the droplet shortcut instead, it will open the folder it represents in Fetch, but will not upload anything.

Droplet shortcuts queue uploads. If an upload to the server referred to by the droplet shortcut is already in progress when you drag items to the droplet shortcut, they will be uploaded after the current upload finishes, instead of opening a new transfer window.

You can also create droplet shortcuts by selecting a folder shortcut in the [Fetch Shortcuts](#) window or another shortcut list window, and choosing **File > Save Droplet Shortcut**.

If the connection to the server opened by a droplet shortcut requires a password, you will be prompted for it when you open the droplet shortcut. If you do not want to be prompted for the password, you need to save the password for the connection in your [keychain](#).

You can copy the droplet shortcut to any location on your Macintosh, or put it in your Dock for easy access. You can also copy the droplet to another Macintosh or send it to other users; however, those users must have a copy of Fetch in order to use the droplet shortcut.

You can choose an [upload format](#) for the droplet shortcut to use by selecting a format from

the **Upload format** menu when creating the droplet. All items dropped on this particular droplet shortcut will always be uploaded in that format. Usually you want **Automatic**, the default choice.

You can also choose to have Fetch quit after an upload started by the droplet shortcut completes by checking the **Quit Fetch after uploading** checkbox when creating the droplet.

If you want to create a double-clickable document that downloads a file, see the [shortcut documents](#) help topic.

If you want to save a list of shortcuts to open later in Fetch, but don't want to automatically open what the shortcuts refer to, you can either [create a shortcut](#) that is added to the Fetch Shortcuts window, or create a [shortcut list document](#) from a [shortcut list window](#).

FTP stands for File Transfer Protocol, and is a set of ground rules (a "protocol") that allows two computers to exchange files over a network. FTP was developed in the 1970s by users of the Advanced Research Projects Agency network, or Arpanet. FTP is one of the oldest official file transfer mechanisms of the Internet (Arpanet's larger successor) and therefore is supported by millions of computers.

Fetch implements one part of FTP, the "client" side. Therefore, it can communicate with any Internet-connected computer that has a "server" program. Some of Fetch's features may not be available on certain servers — this is because the server does not support the feature (for example, not all servers have subfolders) or because the server does not implement the complete FTP specification.

Fetch supports using FTP with TLS/SSL (also known as FTPS) to connect securely to servers.

When you connect to a server using FTP with TLS/SSL, encryption is used to protect the connection between your Macintosh and the server. This protects your password and optionally your data, preventing an eavesdropper from capturing or stealing them as they travel over the network. TLS stands for "Transport Layer Security," and SSL stands for "Secure Sockets Layer." TLS is a newer version of the SSL protocol.

There are three different methods of establishing FTP with TLS/SSL connections; Fetch only supports two of them. The two supported methods are:

- "AUTH TLS," also known as "FTPES," "Explicit SSL," or "Explicit FTPS." This is the preferred method according to the RFC that defines FTP with TLS/SSL.
- "SSL connect," also known as "Implicit SSL" or "Implicit FTPS." This is an older and no longer encouraged method of establishing FTP with TLS/SSL, but it is still somewhat common.

The third method, "AUTH SSL," is also no longer encouraged and is not supported by Fetch.

To connect to a server using FTP with TLS/SSL, choose **FTP with TLS/SSL** from the **Connect using** pop-up menu in the [New Connection dialog](#). When you do this, Fetch will try to connect using Explicit SSL (FTPES) by default.

If you want to use the SSL connect (Implicit SSL) method, you must choose **FTP with TLS/SSL** and also enter 990 in the port field of the New Connection dialog. You may need to click the New Connection dialog's disclosure  button to display the Port field. If you do not enter this specific port number, Fetch will try to establish an FTP with TLS/SSL connection using AUTH TLS.

Choosing **FTP with TLS/SSL** enables the **Enable Encryption** checkbox. If this box is checked, all of your communication will be protected; otherwise only your password will be protected, but performance may be better. Some servers do not support encrypting data when using FTP with TLS/SSL.

To use FTP with TLS/SSL for secure connections in Fetch, the server you are connecting to must support FTP with TLS/SSL connections. Not all servers support FTP with TLS/SSL connections; if you try to connect with FTP with TLS/SSL to a server that doesn't support it, you will receive an error. Your network administrator or service provider can tell you if your server supports FTP with TLS/SSL.

Servers that support FTP with TLS/SSL have a certificate that Fetch uses to verify the security and identity of the server. A server's certificate is issued by a certificate authority, which also vouches for the legitimacy of the certificate by digitally signing it. Your Macintosh has a list of certificate authorities that it trusts, and if the server's certificate is signed by one of those authorities, it will be accepted automatically. However, if the certificate is expired, or not signed by a trusted authority, or has another problem, Fetch will display a warning asking you whether you want to continue connecting to the server or not. For more information about the warning, see the [Verify Certificate dialog](#) help topic.

While both are secure file transfer protocols and have similar names, FTP with TLS/SSL (FTPS) should not be confused with [SFTP](#) (SSH File Transfer Protocol). SFTP is a completely separate protocol from [FTP](#); whereas FTP with TLS/SSL uses the same protocol as FTP, but is wrapped in

an encryption layer.

Fetch supports a special URL format for specifying FTP with TLS/SSL connections; FTP with TLS/SSL URLs start with "ftps://" and have an option at the end to specify whether to use encryption or not. See the [Uniform Resource Locators \(URLs\)](#) help topic for more information.

For more information about security in Fetch, see the [Security](#) help topic.

Gzip (short for GNU zip) format is an open-source compression format common on Unix systems. Gzip compresses a file's data to take up less space. Gzip files typically have names ending with the extension ".gz". Fetch can compress files with gzip when putting files, and automatically decode them when getting files.

Gzip format will only encode the data part of Macintosh files, so you should only use it for files that contain no special Macintosh information. Gzip format may be a good choice if you have large files that compress well and your target audience is Unix users. Fetch's automatic decoding or StuffIt Expander can decode gzip files on the Macintosh.

See the [upload formats](#) help topic for information about encoding files in Gzip format when uploading. You cannot set Gzip format as your default upload format

Fetch automatically recognizes and decodes Gzip files when downloading, although automatic recognition can be turned on and off by selecting a remote file whose name ends with ".gz", choosing **Remote** > **Get Info** to show the [info window](#), and, in the [Transfer Options](#) pane, checking or unchecking the **Automatically decode files like this** checkbox.

Fetch automatically recognizes and decodes Gzip files when downloading, although automatic recognition can be turned on and off by either:

- Selecting a remote file whose name ends with ".gz", choosing **Remote** > **Get Info** to show the [info window](#), and, in the [Transfer Options](#) pane, checking or unchecking the **Automatically decode files like this** checkbox — this will enable or disable decoding of Gzip files but will not affect the decoding of other formats, such as Zip or StuffIt archives;
- Or choosing **Fetch** > **Preferences**, clicking the **Download** tab, and checking or unchecking the **Allow automatic decoding of downloaded files** checkbox — this will enable or disable decoding of all file types.

For more information about using gzip in Fetch, see the [StuffIt and Archive formats](#) help topic. For more information about the gzip format in general, see the [Gzip home page](#).

Related topics:

- [StuffIt and Archive formats](#)
- [Upload formats](#)
- [Transfer Options](#)



## [Fetch Help](#) > [Concepts](#) > Gzipped Tar Archive Format

Gzipped Tar Archive format is a compression and archive format common on Unix systems. It uses Tar, a Unix archiving utility, to combine files into an archive, and then [Gzip](#) to compress the archive to make it smaller. Gzipped Tar archives are sometime known as tarballs. Gzipped Tar archives typically have names ending with the extensions ".tar.gz" or ".tgz". Fetch can create Gzipped Tar archives when putting files, and automatically decode them when getting files.

Gzipped Tar Archive format will only encode the data part of Macintosh files, so you should only use it for files that contain no special Macintosh information. Gzipped Tar Archive format may be a good choice if you want to combine a set of files together in one archive for downloading by Unix users. Fetch's automatic decoding or StuffIt Expander can decode Gzipped Tar archives on the Macintosh.

See the [upload formats](#) help topic for information about encoding files in Gzipped Tar Archive format when uploading. You cannot set Gzipped Tar Archive format as your default upload format.

Fetch automatically recognizes and decodes Gzipped Tar archives when downloading, although automatic recognition can be turned on and off by selecting a remote file whose name ends with ".tar.gz", choosing **Remote** > **Get Info** to show the [info window](#), and, in the [Transfer Options](#) pane, checking or unchecking the **Automatically decode files like this** checkbox.

Fetch automatically recognizes and decodes Gzipped Tar archives when downloading, although automatic recognition can be turned on and off by either:

- Selecting a remote file whose name ends with ".tar.gz", choosing **Remote** > **Get Info** to show the [info window](#), and, in the [Transfer Options](#) pane, checking or unchecking the **Automatically decode files like this** checkbox — this will enable or disable decoding of Gzipped Tar archives but will not affect the decoding of other formats, such as Zip or StuffIt archives;
- Or choosing **Fetch** > **Preferences**, clicking the **Download** tab, and checking or unchecking the **Allow automatic decoding of downloaded files** checkbox — this will enable or disable decoding of all file types.

For more information about using Gzipped Tar archives in Fetch, see the [StuffIt and Archive Formats](#) help topic. For more information about the Gzip format in general, see the [Gzip home page](#).

Related topics:

- [StuffIt and Archive Formats](#)
- [Upload formats](#)
- [Transfer Options](#)

Fetch supports connecting to servers securely using Kerberos via the FTP with GSSAPI and FTP with KClient [connection types](#).

Kerberos is a network security system developed at the Massachusetts Institute of Technology (MIT) that allows two computers on an insecure network to trust each other, and provides ways for the computers to exchange information securely. It is most often used inside companies or colleges, to regulate access to a wide variety of network services — one of the benefits of Kerberos is that a single username and password can grant a user access to many services. Kerberos support is built into Mac OS X, Microsoft Windows, and many UNIX operating systems.

There are two versions of Kerberos in wide use, Kerberos v5 and Kerberos v4. Kerberos v5 is the latest version of Kerberos, and is often accessed via the Generic Security Service API (GSSAPI) programming interface. KClient is a popular interface to Kerberos v4 for the Macintosh. If you have a choice, you should use GSSAPI (Kerberos v5) connections instead of KClient (Kerberos v4) for Kerberos FTP connections, because of the improved security of Kerberos v5. To encourage GSSAPI (Kerberos v5) use, Mac OS X 10.5 and later do not include support for KClient or Kerberos v4, so KClient connections are not available in Fetch on those versions of Mac OS X.

To use Kerberos for secure connections in Fetch, the server you are connecting to must also support Kerberos. If you try to connect with Kerberos to a server that doesn't support it, you will receive an error.

You must also have a proper Kerberos configuration file on your Macintosh (if you are already using other Kerberos services successfully, you probably have one), and have an account in the Kerberos realm that the server uses for authentication. Consult your network administrator or local tech support for further information on using Kerberos at your organization.

To connect to a server using Kerberos, choose either **FTP with GSSAPI** or **FTP with KClient** from the **Connect Using** pop-up menu in the [New Connection dialog](#). Choosing GSSAPI or KClient enables the **Enable Encryption** checkbox. If this box is checked, all of your communication will be protected; otherwise only your password will be protected, but performance may be better. Note also that the password field disappears; your Kerberos ticket will be used to authenticate you instead. Some Kerberos connections may require you to type a password later in the connection sequence; if so, you will be prompted.

Fetch supports a special URL format for specifying Kerberos FTP connections. See the [Uniform Resource Locators \(URLs\)](#) help topic for more information.

Unlike earlier versions, Fetch 5 and later do not require the Mac OS X Kerberos Extras from MIT to use Kerberos with FTP connections.

For more information about Kerberos on the Macintosh, see:

- [Kerberos for Macintosh \(MIT\)](#)

For more general information about Kerberos, see:

- [Kerberos: The Network Authentication Protocol \(MIT\)](#)

For more information about security in Fetch, see the [Security](#) help topic.

Related Fetch help topics:

- [Security Preferences](#)
- [SFTP](#)
- [FTP with TLS/SSL](#)

In addition to the keyboard shortcuts shown next to menu commands, you can use these special keyboard shortcuts to work in Fetch.

### Transfer Windows

Action	Shortcut
Download files or multiple folders	Command-Down arrow
Go to parent folder (go up a level)	Command-Up arrow
Open a single folder	Command-Down arrow or Command-O
Rename item	Return or Enter
Quick Look file	Space bar
Stop transfer	Command-. (Command-period)

### New Connection Dialog

Action	Shortcut
Cycle through shortcuts	Command-] and Command-[ <i>French only: Command-) and Command-(</i>
Cycle through recent connections	Command-Option-] and Command- Option-[ <i>French only: Command-Option-) and Command-Option-(</i>
Choose <b>FTP</b> as connection type	Command-0
Choose <b>SFTP</b> as connection type	Command-1
Choose <b>FTP with TLS/SSL</b> as connection type	Command-2
Choose <b>FTP with KClient</b> as connection type	Command-4
Choose <b>FTP with GSSAPI</b> as connection type	Command-5
Toggle <b>Add to keychain</b> checkbox	Command-K
Toggle <b>Enable encryption</b> checkbox	Command-E

### Fetch Shortcuts / Shortcut List Windows

Action	Shortcut
Download file shortcut	Command-Down arrow
Open folder shortcut	Command-Down arrow
Open New Connection dialog filled out with information from selected shortcut	Command-Option-Down arrow
Rename shortcut	Return or Enter
Quick Look shortcut	Space bar

### New Shortcut/Edit Shortcut Dialog

Action	Shortcut
Choose <b>FTP</b> as connection type	Command-0
Choose <b>SFTP</b> as connection type	Command-1
Choose <b>FTP with TLS/SSL</b> as connection type	Command-2
Choose <b>FTP with KClient</b> as connection type	Command-4

Choose **FTP with GSSAPI** as connection type  
Toggle **Enable encryption** checkbox

Command-5  
Command-E

## Preferences Window

### Action

Cycle through panes

### Shortcut

Command-Shift-[ and Command-Shift-]

## [Fetch Help](#) > [Concepts](#) > **Keychain Support**

Keychains are a password storage facility of Mac OS X that lets you securely store your various passwords in a central place, so that you do not have to remember or retype your passwords all the time. Fetch supports keychains for plain FTP, SFTP, and FTP with TLS/SSL connections.

To save a password in your keychain for future use, check the **Add to keychain** checkbox in the [New Connection dialog](#) or [Mirror Connection dialog](#). When you enter a password for a shortcut in the [Edit Shortcut dialog](#), the password is automatically saved in your keychain.

After you enter a hostname, a username, and choose a connection type in the [New Connection dialog](#), Fetch will check to see if a password matching that information has been saved in your keychain. If one has, it will be automatically entered in the Password field. Also, you will not be prompted for a password when using a [shortcut](#) or [recent connection](#) if the password for that connection is saved in your keychain.

To avoid being prompted for passwords when you use the [Fetch widget](#), a [droplet shortcut](#), or a [mirror document](#), the password for any connections that will be used must be in the keychain.

While previous versions of Fetch allowed you to save a shortcut password in the shortcut itself, now all shortcut passwords are saved in your keychain. When you upgrade to Fetch 5.1 or later, your shortcut passwords will be automatically moved to the keychain if they are not stored there already.

Your keychain is usually automatically unlocked when you start or log in to Mac OS X.

Since Fetch 5.1 and later store passwords for shortcuts in your [keychain](#), moving the shortcuts file from one Macintosh with Fetch 5.1 or later to another will move all the information about your shortcuts except the passwords.

The simplest solution, especially if you just have a few passwords, is to re-enter your passwords on the new Macintosh (where they will be saved in that Macintosh's keychain). Otherwise, see the [moving keychains and passwords to another Macintosh](#) help topic for more options.

For more information about security in Fetch, see the [Security](#) help topic.

MacBinary III (often just referred to as MacBinary) is a format for representing all the information in a Macintosh file (data and resource fork) in one binary file. It is useful for storing a Macintosh file on a non-Macintosh system for later retrieval. Unlike [BinHex](#), MacBinary is a compact format, but it cannot be passed through most electronic mail systems. MacBinary files typically have names ending with the extension ".bin".

Fetch can get and put files in MacBinary III format, as well as get files in MacBinary I and MacBinary II format (earlier versions of the MacBinary standard). See the [upload formats](#) help topic for information about automatically encoding files in MacBinary when uploading.

Fetch automatically recognizes and decodes MacBinary files when downloading, although automatic recognition can be turned on and off by selecting a remote file whose name ends with ".bin", choosing **Remote > Get Info** to show the [info window](#), and, in the [Transfer Options](#) pane, checking or unchecking the **Automatically decode files like this** checkbox.

Fetch automatically recognizes and decodes MacBinary files when downloading, although automatic recognition can be turned on and off by either:

- Selecting a remote file whose name ends with ".bin", choosing **Remote > Get Info** to show the [info window](#), and, in the [Transfer Options](#) pane, checking or unchecking the **Automatically decode files like this** checkbox — this will enable or disable decoding of MacBinary files but will not affect the decoding of other formats, such as Zip or Stuffit archives;
- Or choosing **Fetch > Preferences**, clicking the [Download](#) tab, and checking or unchecking the **Allow automatic decoding of downloaded files** checkbox — this will enable or disable decoding of all file types.

Some FTP servers (especially ones that run on the Macintosh, such as Rumpus and CrushFTP) recognize MacBinary format. If Fetch determines that it is connected to one of these servers, it will try to use MacBinary for all transfers.

Related topics:

- [Upload formats](#)
- [Upload Preferences](#)
- [Transfer Options](#)

A mirror document lets you save the settings from the [Mirror window](#) to use again later. Mirroring lets you automatically copy all the new or changed files from a local Macintosh folder to a server folder, or vice versa.

If you want to mirror the same folders repeatedly, for example to keep a website up to date, a mirror document will save you time because you don't have to set up the Mirror window repeatedly. A mirror document can make uploading the website changes as simple as double-clicking.

Mirror documents can either start mirroring automatically when double-clicked, or just open the Mirror window filled out with the folders specified in the document.

You can create mirror documents with the **Save** button in the [Mirror window](#).

To open a mirror document, either double-click it in the Finder, or choose **File > Open** in Fetch.

If the connections saved in a mirror document require passwords, you will be prompted for them when mirroring starts (either automatically or when you click the **Mirror** button). If you do not want to be prompted for passwords, you need to save the passwords for the connections in your [keychain](#).

For more information about mirroring and creating mirror documents, see the [Mirror window](#) help topic, or the [mirroring a local folder to a remote folder](#) topic.



## [Fetch Help](#) > [Concepts](#) > **One-Time Password**

One-Time Password (OTP) is an Internet standard challenge-response password system. Fetch automatically supports the OTP system, so usually you do not have to do anything special if your server uses OTP.

Servers running OTP software respond to the USER command with special challenge information that is used only once. The FTP client must respond with a password based on the user's secret password and the challenge information. Since the challenge information is never reused, the password that the client sends back changes with each session. That makes it impossible for a wiretapper to collect a password that could then be used to access the account.

When Fetch detects an OTP challenge, it automatically computes the one-time password, based on the secret password entered by the user. This is more convenient than using a separate program to perform the calculation. However, because the process is automatic, it carries a subtle security risk. If the server for some reason does not issue the OTP challenge, Fetch will send the secret password over the network instead of the response derived from the secret password. To be warned in this case, check the **Warn before sending password insecurely** box in the [Security preferences](#) pane.

If, for some reason, you need to compute the response yourself, enter "\*\*\*CHALLENGE\*\*\*" as your secret password in the password field. Fetch will show you the server's challenge and prompt you for a response.

OTP is only supported in FTP connections (and possibly FTP with TLS/SSL connections), not in SFTP connections.

OTP is based on the older [S/Key](#) system developed by Bellcore, which Fetch continues to support.

For more information about security in Fetch, see the [Security](#) help topic.

For more about the One-Time Password system, see [RFC 2289: A One-Time Password System](#).

Fetch shows a small padlock icon in several windows to indicate the security level of a connection. This icon appears in the info bar of [transfer windows](#), the [Mirror window](#), [shortcut list windows](#), and the [Fetch widget](#).

A locked padlock icon indicates a totally secure connection; that is, you've connected with SFTP, FTP with TLS/SSL, or Kerberos with encryption, and both your password and your file transfers will be secure and encrypted.

An unlocked padlock icon indicates a connection is using FTP with TLS/SSL or Kerberos, but encryption is not enabled — your password will be sent securely, but encryption will not be used to protect your file transfers.

No padlock icon is displayed if the connection is not secure at all, that is, it's a regular FTP connection.

The password prompt dialog has a larger lock icon that indicates only the protection of your password. A closed lock indicates that your password will be protected by encryption. An open lock icon indicates that it will not be protected by encryption. This large lock icon says nothing about whether your file transfers will be protected.

Note that not all servers support secure connections.

For information about making secure connections, see the [making secure connections](#) or [New Connection dialog](#) help topics.

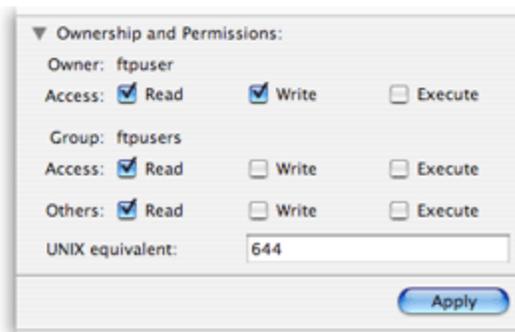
See the [Security](#) help topic for more general information about security in Fetch.

## [Fetch Help](#) > [Concepts](#) > [Permissions](#)

Access to files and folders on servers is usually governed by permission settings. Permissions restrict which users can view, get, edit, put, or delete files, and which users can look inside or make changes to the contents of folders. For instance, you may have permission to look at and download a file on a server (called read permission), but not permission to delete it or make changes (called write permission).

You can view remote items' permissions with the [Get Info](#) command, and if you are the owner of the items, change their permissions. Typically you can only view or set permissions on servers running the UNIX operating system.

The [info window](#) displayed by **Get Info** lets you specify permissions by checking boxes or entering a UNIX equivalent.



Permissions can be specified for three sets of users:

- Owner – A single user, usually the user who created the item or whose home folder it resides in.
- Group – A set of users. Groups are usually created by the server maintainer. Setting the permissions for an item's group gives any member of that group any rights specified by the group permissions.
- Others – All users who can access the server. (Sometimes known as "world" permissions.)

While Fetch can display the owner and group of an item, you cannot change them with Fetch. You also cannot change or view the members of a group using Fetch. For information on changing ownership and group membership, contact your server administrator or hosting provider.

For each set of users, there are three types of permissions that can be specified, allowing the users to:

- Read – View or download files, but not to edit, rename or delete them. Perform some UNIX commands on folders, but not actually view the contents of them (execute permission is required to do that).
- Write – Edit the contents of files. Rename folders, and add or remove items from folders (when the users also have execute permission for the folders).
- Execute – Run programs or scripts when on the server (it doesn't mean much when working with the file in Fetch). See the contents of folders, delete folders, and in conjunction with write permission, add or remove items from folders.

The permissions checkboxes are simultaneously translated into their UNIX equivalent, which is

shown in an edit field below the checkboxes. You can also enter a UNIX equivalent value if you prefer, and the checkboxes will update as soon as you have typed in a valid value. Click the "UNIX permissions" label to switch among the three common UNIX formats (examples: 644, `rw-r--r--`, or `u=rw,g=r,o=r`).

To change the permissions, check the permission boxes as desired, or enter a UNIX equivalent value in the edit field, then click the **Apply** button. See the [viewing and setting permissions](#) topic for a more detailed set of steps.

If you have selected multiple items, some of the checkboxes may have a dash instead of being unchecked or checked. That means not all of the items have the same permissions. Leaving a checkbox with a dash unchanged will preserve the mixed permissions, even if you change other permissions. If you click a checkbox with a dash, you can force the permissions to all become the same.

To discard any changes to the permissions, just close the info window without clicking the **Apply** button.

Setting a remote folder's permissions does not set the permissions of its contents — it only changes the permissions of the folder itself.

You can also view the permissions of multiple items, in UNIX format only, with the [View Verbose File List](#) command.

For just uploaded items on FTP connections, Fetch may show all the permissions checkboxes as unchecked and the owner and group as "unknown". This is not the true state of the permissions. To see the true permissions, you must click the [Refresh](#)  button in the transfer window, and get info on the items again. You can set the permissions immediately without refreshing. On SFTP connections, you will see the true permissions immediately, but you must refresh to see the true owner and group.

You can use the [Send FTP Commands](#) command to [set upload permissions](#), a function supported by some servers that lets you specify the permissions to be used for files that are yet to be uploaded.

A script for setting permissions of the contents of a folder recursively is available as part of the example Fetch AppleScripts, which you can download from the [Fetch Softworks website](#). You can also easily set up an Automator workflow to recursively set permissions. See the [more examples of using Automator with Fetch](#) help topic for an example.

Related topics:

- [Get Info](#)
- [Info window](#)
- [View Verbose File List](#)

S/Key is a challenge–response one–time password system developed by Bellcore. Fetch automatically supports the S/Key system, so usually you do not have to do anything special if your server uses S/Key.

Servers running the S/Key software respond to the USER command with special challenge information that is used only once. The FTP client must respond with a password based on the user's secret password and the challenge information. Since the challenge information is never reused, the password that the client sends back changes with each session. That makes it impossible for a wiretapper to collect a password that could then be used to access the account.

When Fetch detects an S/Key challenge, it automatically computes the one–time password, based on the secret password entered by the user. This is more convenient than using a separate program to perform the calculation. However, because the process is automatic, it carries a subtle security risk. If the server for some reason does not issue the S/Key challenge, Fetch will send the secret password over the network instead of the response derived from the secret password. To be warned in this case, check the **Warn before sending password insecurely** box in the [Security preferences](#) pane.

If for some reason you need to compute the response yourself, enter "\*\*\*CHALLENGE\*\*\*" as your secret password. Fetch will show you the server's challenge and prompt you for a response.

While the [S/Key](#) standard says that challenge responses should be hashed with MD4, some locations use MD5 for hashes instead. If your site/server is one of those, check the **Use MD5 for S/Key** preference in the [Obscure Preferences](#) pane so that your responses to challenges will work properly.

Fetch also supports the [One–Time Password \(OTP\)](#) system, which supersedes S/Key. See the [One–Time Password](#) topic for more information.

S/Key is only supported in FTP connections (and possibly FTP with TLS/SSL connections), not in SFTP connections.

Related topics:

- [One–Time Password](#)

Fetch supports using SFTP to connect securely to servers.

SFTP stands for "SSH File Transfer Protocol." When you connect to a server using SFTP, SSH encryption is used to protect the connection between your Macintosh and the server. This protects your password and your data, preventing an eavesdropper from capturing or stealing them as they travel over the network.

Despite the similarity in name and operation, SFTP is a completely different protocol from FTP and does not support all the same features and commands as FTP. Also, while they are both secure file transfer protocols and have similar names, FTPS ([FTP with TLS/SSL](#)) should not be confused with SFTP.

To use SFTP for secure connections in Fetch, the server you are connecting to must also support SFTP. If you try to connect with SFTP to a server that doesn't support it, you will receive an error. Your network administrator or service provider can tell you if your server supports SFTP, and what other information you might need to use SFTP if it does.

To connect to a server using SFTP, choose **SFTP** from the **Connect using** pop-up menu in the [New Connection dialog](#). Notice that the **Enable Encryption** checkbox becomes checked and disabled. SFTP always uses encryption — unlike FTP with TLS/SSL or Kerberos, where encryption can be turned off.

Some other differences between SFTP and FTP connections worth noting are:

- When uploading text files via SFTP, line endings are not translated as they are when uploading text files via FTP. Instead the line ending of the file is preserved as whatever it was on the Macintosh.
- SFTP does not have [active and passive](#) connection modes, and thus works better with firewalls, routers, and NAT devices.

For more information about security in Fetch, see the [Security](#) help topic.

Related topics:

- [Security Preferences](#)
- [Kerberos](#)

Fetch supports a number of Apple events. This makes it possible to automate tasks in Fetch with a scripting tool such as AppleScript. It also makes it possible for other programs to use Fetch as a helper application for resolving FTP and SFTP Uniform Resource Locators (URLs). This help topic describes Fetch's support for the URL, Fetch, and Interarchy Apple event suites.

Fetch also provides a variety of Automator actions. You may find that setting up an Automator workflow to automate a Fetch task is even easier than using AppleScript. For more information, see the [introduction to using Automator with Fetch](#) and [Automator actions](#) help topics.

Fetch supports Apple event recording. The easiest way to get a feel for scripting Fetch, or to find out how to script a task, is to turn on recording in your script editor, perform the task manually, and then examine the script that has been recorded. Most user actions will be recorded as script statements. For security reasons, Fetch will not record passwords, so when you run the recorded scripts you may have to furnish passwords, unless those passwords are stored in your [keychain](#).

It is also useful to examine Fetch's Apple event dictionary, since it provides a complete list of the supported object classes, properties, and events.

You can download a set of example AppleScripts for Fetch from the [Fetch Softworks website](#).

If you used AppleScript with previous versions of Fetch, you should read the [AppleScript changes](#) help topic for more information about changes to Fetch's scripting terminology.

## URL Suite

Fetch supports the **geturl** (GURL) event from the Standard URL Apple event suite, originally designed by John Norstad of Northwestern University and a group of Internet software authors. In AppleScript, the event is specified as follows:

```
tell application "Fetch"
    geturl "ftp://ftp.fetchsoftworks.com/"
end tell
```

The **geturl** event is used by the Mac OS and other applications to pass URLs to other applications for resolution. So email programs, web browsers, and newsreaders can pass FTP and SFTP URLs to Fetch using **geturl**.

## Fetch Suite

Fetch supports the following object classes that are specific to Fetch:

- the **application** object, which can contain each of the following kinds of objects:
  - **window** objects
  - **transfer window** objects, which can contain:
    - **remote item** objects
    - **remote file** objects
    - **remote folder** objects
  - **shortcut list window** objects, which can contain:
    - **shortcut** objects

the **fetch shortcuts window** object, which can contain:

- **shortcut** objects
- **text window** objects
- **media window** objects
- the **transcript window** object
- **url** objects

Fetch objects also have properties specific to Fetch. For example, the application object has all the preferences settings as properties, so you can refer to its **download folder** or **text line ending style**.

You can use the standard Core suite of commands to manipulate Fetch objects. The standard commands include **close**, **count**, **delete**, **duplicate**, **exists**, **get**, **make**, **move**, **open**, **quit**, **save**, and **set**. To this list Fetch adds the following commands: **download**, **view verbose file list**, **put into**, **send to**, **edit**, **edit new text file**, **get information about**, **mirror**, **refresh file list**, **stop**, **view verbose file list**, and **view** (10.4 only).

Here are examples of some of these events, used with a variety of Fetch Apple event objects:

```
close text window "ftp.fetchsoftworks.com messages"
set bcount to count shortcut list window "Fetch Shortcuts" each shortcut
delete remote file "a.out" of transfer window "csvax.polytech.edu"
duplicate remote file "cool.gif" to alias "Hobbes:Desktop Folder:"
set fexists to exists shortcut "cool site" of shortcut list window "Fetch Shortcuts"
get display server messages
make new transfer window with properties {hostname:"ftp.fetchsoftworks.com", initial
folder:"/example"}
move shortcut 1 of shortcut list window "Fetch Shortcuts" to shortcut list window "Best
Shortcuts"
open parent folder of transfer window 1
quit
save text window "ftp.fetchsoftworks.com messages" in file "Hobbes:server messages"
set transfer mode of transfer window 1 to Binary
download url "ftp://ftp.fetchsoftworks.com/pub/mac/README"
view verbose file list transfer window "ftp.fetchsoftworks.com"
put into transfer window "ftp.fetchsoftworks.com" item alias "Hobbes:Desktop Folder:Picture 3"
binary format MacBinaryIII
send to url "ftp://ftp.fetchsoftworks.com/" ftp command "LIST -IR"
```

## Interarchy Suite

Nolobe's Interarchy, originally known as Anarchie, was the first Macintosh FTP client to support scripting via Apple events. Fetch supports a subset of the Interarchy suite of events, to make it easier to reuse scripts written for Interarchy and Anarchie (and because it is a straight-forward and functional collection of events). The supported events are:

To download a remote file:

```
fetch alias "My hard disk:my folder:cool file" host "ftp.example.com" user "fred" password
"secret" path "remotedir/cool.file.hqx"
```

To upload a file:

```
store alias "My hard disk:my folder:file to upload" host "ftp.example.com" user "fred" password
"secret" path "uploaddir/new.file"
```

To delete a remote file or remote folder:

```
remove host "ftp.example.com" user "fred" password "secret" path "remotedir/bad.file"
```

To view a file list:

```
interarchy list host "ftp.example.com" user "fred" password "secret" path "remotedir"
```

To create a remote folder:

```
mkdir host "ftp.example.com" user "fred" password "secret" path "remotedir/newdir"
```

To rename a remote file or remote folder:

```
rename host "ftp.example.com" user "fred" password "secret" path "remotedir/old.name"  
newname "new.name"
```

To edit a remote file with BBEedit (or other selected text editor):

```
interarchy edit host "ftp.example.com" user "fred" password "secret" path  
"remotedir/filttoedit.txt"
```

To send a raw FTP command:

```
sendcommand host "ftp.example.com" user "fred" password "secret" command "SITE A LIST"
```

The **user** and **password** parameters for any of the Interarchy Suite commands can be omitted if you are using anonymous FTP. You cannot omit the **path** parameter when using the forms above. Also, the **host**, **user**, **password**, and **path** parameters can be replaced by a single **url** parameter; for example:

```
fetch alias "My hard disk:my folder:cool file" url  
"ftp://fred:secret@ftp.example.com/remotedir/cool.file.hqx"
```

If you want to use SFTP instead of FTP with any of the Interarchy suite commands, append "**sftp true**" to the command, for example:

```
fetch alias "My hard disk:my folder:cool file" host "ftp.example.com" user "fred" password  
"secret" path "remotedir/cool.file.hqx" sftp true
```

Fetch includes support for several types of security when making connections: SFTP, FTP with TLS/SSL (FTPS), and Kerberos. It also supports the Mac OS [keychain](#) for securely storing passwords locally.

The two goals of the security features in Fetch are protecting the privacy and integrity of your Fetch sessions. Protecting privacy means making it harder for anyone to listen in on your Fetch sessions and watch your password and file transfers go by. Protecting integrity means making it more difficult for an attacker to alter your Fetch sessions somewhere between your computer and the server, for the purpose of corrupting data or gaining unauthorized access. Depending on the nature of your use of Fetch, and the sensitivity of your account and data, these features may be anything from pointless (you are on a private network of trusted users, exchanging innocuous data) to vital (you are moving classified files over a public network).

When connecting to a server that supports secure connections, Fetch protects the privacy and integrity of its sessions by arranging with the server to encrypt all the information that it sends over the network. This involves scrambling the information so that it can only be unscrambled by someone with a secret key, which is known to only your Macintosh and the server.

When you connect to a server without security (by choosing the plain FTP [connection type](#)), you are said to be sending data, such as your password, in the clear — which means the data is not protected or encrypted in any way. However, plain FTP does not require any special server support, and remains common.

Secure connections come in several types: SFTP, FTP with TLS/SSL, and Kerberos (aka GSSAPI and KClient). Not all servers support secure connections, and a particular server may only support one kind of secure connection. Your network administrator or service provider can tell you if your server supports secure connections, and what other information you might need.

Encrypting data does take some time, and may slow down sessions depending on the speed of your Macintosh and the load on the server. For that reason, Fetch makes it possible to turn encryption off when using FTP with TLS/SSL or Kerberos, in case speed is a higher priority than privacy. (Encryption is always on in SFTP connections; you cannot disable it.) But even if encryption is disabled, passwords are exchanged securely in order to keep them from being captured or seen on the network. It is only your file transfers that are sent unencrypted when encryption is disabled.

Several Fetch windows display a small padlock icon. A locked padlock icon indicates a totally secure connection, that is, you've connected with SFTP, FTP with TLS/SSL, or Kerberos with encryption, and both your password and your file transfers will be secure and encrypted. An unlocked padlock icon indicates a connection that uses FTP with TLS/SSL or Kerberos but encryption is not enabled — your password is sent securely, but encryption will not be used to protect your transfers. No padlock icon is displayed if the connection is not secure at all, that is, it's a plain FTP connection. The padlock icon appears in the info bar of transfer windows, the Mirror window, shortcut list windows, and the Fetch widget.

For information about making secure connections, see the [making secure connections](#) or [New Connection dialog](#) help topics.

Fetch also supports [One-Time Password](#) and [S/Key](#), which are challenge-response password systems that protect account passwords, but do not encrypt transferred data.

When you open an FTP, SFTP, or FTP with TLS/SSL connection, you can store the password for

your account in your Mac OS keychain. The keychain lets you store your various passwords securely in a central place so that you don't have to remember all your different passwords. Fetch will also look in your keychain for your password to a server when you use a shortcut, open a recent connection, use the Fetch widget, or open a mirror document. See the [keychain](#) help topic for more information.

Other topics related to security in Fetch are:

- enabling and disabling encryption with the [Enable Encryption](#) command;
- the [Security Preferences](#) pane, where you can set warnings and options for secure FTP connections;
- more information about [SFTP](#), [FTP with TLS/SSL](#), and [Kerberos](#);
- specifying a secure connection type in a [shortcut](#).

## [Fetch Help](#) > [Concepts](#) > Server-to-Server Transfers

Fetch supports transferring files and folders from one server to another.

To transfer a file or folder from one server to another, connect to both servers and drag the items from one file list to another. You can stop the transfer by clicking the **Stop**  button in either window. It is also possible to synchronize two folders on two different servers using the **Remote** > **Mirror** command — see the [Mirror](#) help topic for more information.

Normally Fetch does a server-to-server transfer by transferring the data from one server to your Macintosh, and then immediately from your Macintosh to another server. If the servers in question support it, and encryption is not being used, the transfer can take place directly between the two servers (known as FXP). This can offer dramatic performance advantages if the connection between the two servers is much faster than your connection to one or the other of them, but unfortunately it can be less reliable and not all servers support it. You can tell Fetch to try direct transfers by checking the **Use direct server-to-server transfers** preference in the [Obscure Preferences](#) pane.

The [Duplicate](#) command in the [file list contextual menu](#) is a special case of a server-to-server transfers, where Fetch opens a second connection to a server and transfers the file from the server to itself.

It's also possible to make a new copy of a file in one folder by Option-dragging the file to a folder icon in the file list. The copy  cursor indicates that the file will be copied rather than simply moved. Like the **Duplicate** command, this will open a second connection and transfer window to the server.

You can also use the **Copy** and **Paste** commands to copy files and folders from one folder to another on the same server, or from one server to another. See the [copying files using copy and paste](#) help topic for more information.

Shortcut documents are files that you can double-click in the Finder to tell Fetch to automatically download remote files or open remote folders.

While using a shortcut document is very similar to using [shortcuts](#) in Fetch, a shortcut document may save you more steps or may suit your workflow better. For instance, if there's a file you download repeatedly because someone you work with updates it regularly, you could create a shortcut document that refers to that file. Then you can just double-click the shortcut document in the Finder, enter your password if necessary, and Fetch will connect to the server where the file is stored and download the file.

If there's a remote folder you always connect to, you can instead create a [droplet shortcut](#) that refers to that folder. Then just double-click the droplet shortcut, enter a password if necessary, and you're there. No navigating is needed in either case.

You create shortcut documents using the [Save Shortcut](#) command, and droplet shortcuts with the [Save Droplet Shortcut](#) command.

To create a shortcut document or droplet shortcut:

1. Open a connection to the server where the file or folder is stored.
2. In the transfer window file list, select the file or folder by clicking it once.
3. Choose **File > Save Shortcut** (for a file) or **File > Save Droplet Shortcut** (for a folder).
4. Enter a name and choose a location for the shortcut document or droplet shortcut.

You can also select multiple files or folders and save a single shortcut document to all of them. When you open that shortcut document, all the files will be downloaded and all the folders will be opened.

To open a shortcut document, either double-click it in the Finder, or choose **File > Open** in Fetch and select the document. To open a droplet shortcut, double-click it in the Finder.

An additional feature of droplet shortcuts is that you can upload files or folders to the remote folder the droplet shortcut represents by dragging and dropping the files to the droplet shortcut's icon. You cannot drag and drop files to shortcut documents, because usually they represent remote files, and you cannot upload files to a file. For more information, see the [droplet shortcut](#) topic.

If the connection to the server opened by a shortcut document requires a password, you will be prompted for it when you open the shortcut document. If you do not want to be prompted for the password, you need to save the password for the connection in your [keychain](#).

If you want to save a list of shortcuts to open later in Fetch, but don't want to automatically download or open what the shortcuts refer to, you can either [create a shortcut](#) that is added to the [Fetch Shortcuts](#) window, or create a [shortcut list document](#) from a [shortcut list window](#).



## [Fetch Help](#) > [Concepts](#) > **Shortcut List Documents**

Shortcut list documents are lists of [shortcuts](#) that you can save and use to recreate a [shortcut list window](#) in Fetch. If you have a set of shortcuts that you'd like to open again later, but want to keep them separate from the list of shortcuts in the [Fetch Shortcuts](#) window, you need to save them in a shortcut list document.

Unlike [shortcut documents](#), shortcut list documents do not download files or open folders when you open them. They store a catalog of shortcuts that you can add to, edit, and remove from later on. Once you've opened the shortcut list document, you can use the shortcuts in it to download files, open folders, or open connections.

Shortcut list windows are just like the Fetch Shortcuts window, except they are not automatically saved or automatically reopened. You must save a shortcut list window as a shortcut list document if you want to use it again after you quit Fetch.

To create a shortcut list document:

1. Create a new shortcut list window with **Shortcuts > New Shortcut List**.
2. Add shortcuts to the shortcut list window.
3. Choose **File > Save**.
4. Enter a name and choose a location for the shortcut list document.

To open a shortcut list document, either double-click it in the Finder, or choose **File > Open** in Fetch and select the shortcut list document.

For more information on working with shortcuts or creating shortcuts, see the [shortcuts](#) and [creating a shortcut](#) help topics.

Related topics:

- [Shortcut documents](#)
- [Save Shortcut](#)
- [Save Droplet Shortcut](#)

Shortcuts are a quick way to access files or folders that are available over the Internet, similar to bookmarks or favorites in web browsers. They let you access items without typing in connection information each time.

A shortcut contains the following information: a name for the shortcut, so you can identify it later; the kind of shortcut it is; the hostname of the server that the shortcut refers to; the username for your account on the server that the shortcut refers to (if necessary); the password for that account (if necessary); the [connection type](#) to use for accessing the server; the path to the remote folder or file (if desired); and some other options such as a special port to use or whether to use encryption when accessing the server.

There are three kinds of shortcuts: file, folder, or unknown. A file shortcut refers to a single file. A folder shortcut refers either to a remote folder or to a server. A shortcut whose kind is unknown indicates that Fetch could not determine if the shortcut refers to a file or a folder.

Typically shortcuts are stored in the Fetch Shortcuts window. This window is automatically saved (in your Preferences folder) and automatically reopened when you start Fetch. To display the Fetch Shortcuts window, choose **Shortcuts > Show Shortcuts**. The shortcuts in the Fetch Shortcuts window can be accessed from several locations in Fetch, such as the **Shortcuts** menu and the New Connection dialog. See the [Fetch Shortcuts](#) help topic for more information.

To create a new shortcut, use the [Make Shortcut](#) command in the Shortcuts pop-up menu of the [New Connection dialog](#), or choose **Shortcuts > New Shortcut**, which displays the [New Shortcut dialog](#). See the [creating a shortcut](#) help topic for detailed steps.

To make the New Connection dialog always open filled with information from a shortcut, set a shortcut to be the default shortcut. You can do this using the **Default shortcut** preference in the [General Preferences](#) pane, or you can create a default shortcut using the **Make Shortcut** command in the New Connection dialog. See the [making Fetch start with information for your server or website](#) help topic for more information.

To delete a shortcut, select the shortcut in the Fetch Shortcuts window and choose **Edit > Delete** or press the Delete key. To change a shortcut, select the shortcut in the Fetch Shortcuts window and choose **Shortcuts > Edit Shortcut**. For more information, see the [deleting or changing a shortcut](#) help topic.

When you open a connection that matches one of your shortcuts, the name of the shortcut is displayed in the title bar of the [transfer window](#).

You can also create lists of shortcuts separate from the Fetch Shortcuts window. These are called shortcut list windows. Unlike the Fetch Shortcuts window, they are not automatically saved and reopened. To create a shortcut list window, choose **Shortcuts > New Shortcut List**. See the [shortcut list window](#) help topic for more information.

Two other shortcut-related items are droplet shortcuts and shortcut documents.

[Droplet shortcuts](#) are used to access remote folders from the Finder or the Dock. You can drag files to a droplet shortcut to have Fetch upload those files, or double-click it to open a remote folder in Fetch. To create a droplet shortcut, select a folder in a transfer window or a shortcut in the Fetch Shortcuts window, and then choose **File > Save Droplet Shortcut**. See the [Save Droplet Shortcut](#) help topic for more information.

A [shortcut document](#) is a document that refers to a file on a server, and will download the file when you double-click on the shortcut document. For example, if you download a file frequently, you can create a shortcut document referring to that file and then just double-click on the shortcut document in the Finder or Dock to accomplish the task. To create a shortcut document, select a remote file in a transfer window, then choose **File > Save Shortcut**. See the [shortcut documents](#) help topic for more information.

Passwords for shortcuts are stored securely in your [keychain](#).

Related topics:

- [New Shortcut / Edit Shortcut dialog](#)
- [Fetch Shortcuts](#)
- [Shortcut list windows](#)
- [Shortcuts menu](#)
- [Shortcut documents](#)
- [Save Droplet Shortcut](#)

 [Fetch Help](#) > [Concepts](#) > **Software Update**

You can be sure you have the latest version of Fetch with the automatic check for update feature or the manual **Check for Update** command.

Normally Fetch periodically checks with Fetch Softworks' web server to see if a newer version of Fetch is available, so you don't have to think about doing it yourself. When Fetch detects that an update is available, it will display a [dialog](#) with information about the update and an option to download and install it.

No personal information is sent when checking for updates. See the [Software Update Privacy](#) help topic for more information.

You can disable the automatic checks by unchecking the **Automatically check for updates** preference in the [Miscellaneous Preferences](#) pane.

If you have disabled the automatic update checks, or if you skipped over or delayed installing an update, you can use the manual [Check for Update](#) command in the **Fetch** menu to see if you're up to date.

You can also download updates from the [Fetch Softworks](#) website.



[Fetch Help](#) > [Concepts](#) > **Software Update Privacy**

Fetch periodically checks with the Fetch Softworks web server to see if a newer version of Fetch is available. Your privacy is protected during these checks — no personal information is sent or disclosed by them.

When Fetch checks for updates, the following information is sent: the version of Fetch you are using, the version of Mac OS X that you are using, your current preferred language (as set in the **International** pane of **System Preferences**), and whether or not your copy of Fetch is registered.

This information is only used to determine if you need to be notified of a newer version of Fetch, and to guide the future development of Fetch. It is not personalized or disclosed.

The web server will also log the date, time, and originating IP address of the check. This information is not used or disclosed.

You can disable the automatic checks by unchecking the **Automatically check for updates** preference in the [Miscellaneous Preferences](#) pane. You can check for updates manually by choosing the [Check for Update](#) command from the **Fetch** menu.

For more general information about the software update process, see the [Software Update](#) help topic.

Stuffit X Archive format is a compression and archive format created by [Allume Systems](#) (formerly Aladdin Systems) as part of their Stuffit line of products, and is a common method for storing files on the Macintosh. The Stuffit X Archive format takes multiple files and encodes them into a single file, compressing the data to take up less space at the same time. Stuffit X archives typically have names ending with the extension ".sitx". Fetch can create Stuffit X archives when putting files, and automatically decode them when getting files.

Stuffit X Archive format will preserve both the data and resource forks of Macintosh files, as well as long file names and other Macintosh Finder information, making it a good format to use if you are creating archives of files to give to other Macintosh users. Stuffit X is an improved version of the original Stuffit (.sit) format — while Fetch only creates Stuffit X files when uploading, it can automatically decode and recognize both Stuffit X and the older Stuffit archive formats when downloading. Even if other Macintosh users are not using Fetch to download files you upload, they can use the free Stuffit Expander to decode and decompress Stuffit X archives. Stuffit Expander is included with some versions of Mac OS X, or is available as a [free download](#); it is also available as a free download for the Windows operating system.

See the [upload formats](#) help topic for information about encoding files in Stuffit X Archive format when uploading. You cannot set Stuffit X Archive format as your default upload format.

Fetch automatically recognizes and decodes Stuffit X archives when downloading, although automatic recognition can be turned on and off by either:

- Selecting a remote file whose name ends with ".sitx" or ".sit", choosing **Remote > Get Info** to show the [info window](#), and, in the [Transfer Options](#) pane, checking or unchecking the **Automatically decode files like this** checkbox — this will enable or disable decoding of Stuffit X archives but will not affect the decoding of other formats, such as Zip archives;
- Or choosing **Fetch > Preferences**, clicking the **Download** tab, and checking or unchecking the **Allow automatic decoding of downloaded files** checkbox — this will enable or disable decoding of all file types.

For more information about using Stuffit X archives in Fetch, see the [Stuffit and Archive formats](#) help topic. For more details about the Stuffit X format and compression, visit the [Allume Stuffit X](#) webpage.

Related topics:

- [Stuffit and Archive formats](#)
- [Upload formats](#)
- [Transfer Options](#)

Fetch supports uploading files in Stuffit, Zip, and other archive formats, and can automatically decode files in Stuffit, Zip and other archive formats after the archives have been downloaded. The archive formats are: [Stuffit X Archive](#) (common on the Macintosh), [Zip Archive](#) (common on Macintosh and Windows), and [Gzipped Tar Archive](#) (common on Unix).

Stuffit support is built into Fetch, so you do not need to have Stuffit installed on your Macintosh to create or decode files in the archive and compression formats.

Uploading files in an archive format can be handy if you want to create a single package of files for other users to download, instead of asking them to download all the files separately. When you upload one or more files and folders with one of the archive formats, all the files are combined together into a single file. Since archives are also compressed, you may save space on the server by placing your files in an archive. Fetch also supports uploading files with [Gzip](#) compression, but Gzip is not an archive format — when you upload multiple files with Gzip, you end up with multiple compressed files.

You can choose to use one of the archive formats when selecting files for uploading in the [Put](#) dialog, or set one as the upload format for a current connection with the [Format](#) menu. You cannot choose an archive format as your default upload format for all new windows.

Being able to decode Stuffit and other archive formats automatically after download eliminates the need to open a separate program, such as Stuffit Expander, to use the files after you've downloaded the archive. Fetch supports automatic decoding of all the formats available as [upload formats](#) — Stuffit X archives (.sitx), Classic Stuffit archives (.sit), Zip archives (.zip), Gzipped Tar archives (.tar.gz, .tgz), BinHex (.hqx) and MacBinary (.bin) encoded files.

Automatic decoding of compression and archive formats can be enabled and disabled for all files by checking or unchecking the **Allow automatic decoding of downloaded files** checkbox in the [Download Preferences](#) pane. Note that disabling decoding of all files will disable Fetch's automatic decoding of BinHex, MacBinary, and AppleSingle files in addition to disabling Stuffit and Zip archive expanding.

Or you can enable or disable automatic decoding of just certain kinds of files in the [Transfer Options](#) pane of the [info window](#). Find a file of the type desired on a server, select it, choose **Remote > Get Info** to show the info window, and in the Transfer Options pane, check or uncheck the **Automatically decode files like this** checkbox.

Even if other users are not using Fetch to download files you upload, the Stuffit and Zip formats are standard and most computers have tools to decode them (such as Stuffit Expander).

In [Automatic upload format](#), files that have both resource and data forks will be uploaded in Zip Archive format, in order to preserve all their Macintosh characteristics.

When Fetch is using Stuffit to encode or decode files on upload or download, a Stuffit progress window will appear, displaying the progress of the encoding or decoding. This progress is separate from the transfer progress displayed in the Fetch transfer window; if you are uploading and encoding a large file, you will first see the Stuffit encoding progress window for a while, and then the Fetch uploading progress bar.

Related topics:

Upload formats

- Transfer Options



The Text format assumes that a file only consists of letters, numbers, and punctuation, such as HTML files and files created by text editors such as BBEdit. Most word processors can create text files as well. Text format is sometimes referred to as ASCII format.

Text format only transfers the data fork of a file — any special Macintosh information stored in the resource fork of a file will not be transferred. Also, when a file is uploaded using Text format over FTP (but not SFTP) connections, line ending conversion is performed. Different computer systems use different characters to denote the end of a line, and Fetch has to convert the line endings of Macintosh files to something the server will understand. You do not want to do this to binary files, such as JPEGs, because that will scramble the data that makes up the picture.

Note: No line ending conversion is performed when uploading over SFTP connections. The line endings of text files after being uploaded to a server using SFTP will be the same as they were on the Macintosh.

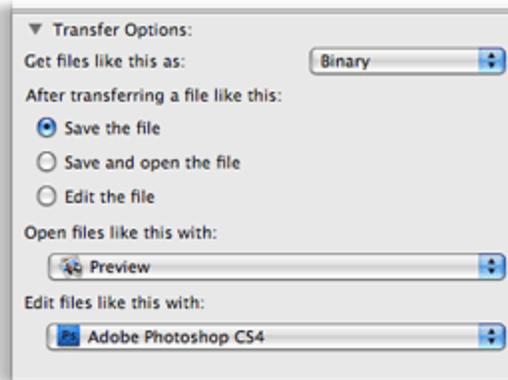
The Text format is mostly useful if you have text files whose names end in an extension usually associated with binary files (such as ending in "bin", "Z", "sit", or "gif"), or for text files that are not identified as text files on your Macintosh regardless of the filename. In most cases, using the [Automatic upload format](#) will choose the correct format for your files and you do not have to worry about choosing a specific one.

There is also a Text download mode for ensuring text files are downloaded correctly if the Automatic download mode does not work correctly. See the [download modes](#) help topic for more information.

Related topics:

- [Upload formats](#)
- [Upload Preferences](#)
- [Download modes](#)

The Transfer Options pane of the [info window](#) lets you tell Fetch how it should download a file, and what it should do with it afterwards.



While Fetch provides good default transfer options for the most common kinds of files, you may find that files you work with do not have the default settings you want, or you may want certain kinds of files to open automatically after downloading (Fetch's initial settings are to just download everything). The Transfer Options pane lets you make these changes.

You can view a remote file's transfer options by selecting a file of the kind you'd like to make changes to and then choosing **Remote** > [Get Info](#), which displays the info window. Note that the Transfer Options pane is not displayed when you select a folder, an alias, or multiple files.

Changes to a file's transfer options apply to every file that has the same filename extension as the one you've selected. For example, if you select a file with an extension of ".doc", the new transfer options you set apply to every remote file with an extension of ".doc". If you select a file that doesn't have an extension, changes you make apply to all files that don't have extensions.

Changes to transfer options take place immediately.

The transfer options are detailed below.

**Get files like this as** – Specifies how this kind of file will be downloaded when the Automatic [download mode](#) is used. If the Automatic download mode seems to download a certain kind of file incorrectly, you may need to change this setting.

**After transferring a file like this** – Specifies what Fetch will do after downloading this kind of file. The choices are:

**Save the file** – Take no further action after downloading files like this.

**Save and open the file** – After downloading, open files like this in the application specified by the **Open files like this with** pop-up menu.

**Edit the file** – When you double-click on files like this, instead of downloading them to your download folder, Fetch will download them to a temporary location, open them in the application specified by the **Edit files like this with** pop-up menu, and changes will be saved back to the server when you save the files in the editing application. See the [Edit](#) command for more information.

**Automatically decode files like this** – This option only appears for certain kinds of encoded

files, such as BinHex, MacBinary, and the [archive and compression formats](#) supported by Fetch, such as Zip archives and Stuffit X archives. When this box is checked, Fetch will automatically decode or expand this kind of file after it is downloaded. The original file is not saved, only the decoded file. When this box is unchecked, Fetch will just save the original file and not attempt to process it further.

If you have trouble downloading an encoded file, uncheck the **Automatically decode files like this** option, download the file again, and try decoding the file with Stuffit Expander or another utility. Automatic decoding may not apply if you have also selected the **Save and open the file** or **Edit the file** option for this kind of file.

You can disable automatic decoding for all files by unchecking the **Allow automatic decoding of downloaded files** checkbox in the [Download Preferences](#) pane. When the **Allow automatic decoding of downloaded files** preference is unchecked, then the **Automatically decode files like this** checkbox in the info window will be disabled.

**Open files like this with** – Choose the application that will open files like this when you double-click them in the Mac OS Finder. This application will also be used to open files when the **Save and open the file** transfer option is set. Changing this option will usually change the icon Fetch displays for this kind of file to reflect the application you've chosen. If the application you want isn't listed, you can select another by choosing the **Other** menu item from this pop-up menu. If Fetch does not remember the application you chose, see the [fixing problems choosing editors and helper applications](#) help topic.

**Edit files like this with** – Choose the application to use for editing files like this with the [Edit](#) command. This application will also be used to edit files when the **Edit the file** transfer option is set. If the application you want isn't listed, you can select another by choosing the **Other** menu item from this pop-up menu. If Fetch does not remember the application you chose, see the [fixing problems choosing editors and helper applications](#) help topic.

For more information on using the transfer options, see the following help topics:

- [Changing the application a file opens with](#)
- [Opening files automatically after downloading](#)
- [Changing the editor for a file](#)
- [Editing files automatically instead of downloading](#)
- [Decoding files automatically](#)

Related topics:

- [Get Info](#)
- [Info window](#)
- [Download modes](#)
- [Edit](#)

## [Fetch Help](#) > [Concepts](#) > **Uniform Resource Locators (URLs)**

Uniform Resource Locators (URLs) are addresses for resources on the Internet: they specify what kind of resource the item is, what server it can be found on, and its name on that server. In the case of files and folders on FTP servers, an FTP URL has the following format:

```
ftp://<username>:<password>@<hostname>:<port>/<path>
```

If the username and password are omitted, the URL is assumed to refer to an anonymous FTP server. The port value is also optional. For example, the URL for a copy of Fetch on Fetch Softworks' FTP server would look like this:

```
ftp://ftp.fetchsoftworks.com/Fetch_5.dmg
```

If you omit the password from a URL, Fetch will prompt you for the password when necessary.

Fetch also supports URLs for SFTP, FTP with TLS/SSL, and Kerberos FTP connections. In Fetch, these kinds of URLs are known as Fetch addresses. When [WebView](#) is set up, Fetch can also create and understand web addresses, which are HTTP URLs that are used to refer to files and webpages in web browsers. See below for more information about these URLs.

URLs were originally designed for use in the World Wide Web (WWW). Fetch uses URLs as part of its [shortcuts](#), and can also use URLs in the [New Connection](#), [Edit Shortcut](#), and [Get](#) dialog. The **Edit > Copy Fetch Address** command copies an FTP URL for items selected in the file list or for the current folder to the clipboard. When WebView is set up for a server, the **Edit > Copy Web Address** command copies an HTTP URL for the selected items. Dragging files or folders from a file list to a text editor inserts a URL for those items. You can create shortcuts in [Fetch Shortcuts](#) or any shortcut list window by pasting or dragging a URL into the window.

Most Macintosh Internet programs can be used as URL helper applications. When you click on a link or URL in another program, it passes the URL to the appropriate helper application, which then retrieves the desired data or performs the desired action.

Fetch can be used as a helper for FTP, FTP with TLS/SSL, and SFTP URLs. To set Fetch as your default helper, use the **Default FTP application** and **Default SFTP application** pop-up menus in the [General Preferences](#) pane.

### **Other URL Formats**

[SFTP](#) URLs are similar to FTP URLs, except they start with "sftp://".

[FTP with TLS/SSL \(FTPS\)](#) URLs are also similar to FTP URLs, except they start with "ftps://", and Fetch supports extensions to FTPS URLs in order to specify whether the connection should use encryption or not. An FTPS URL has the following format:

```
ftps://<username>@<hostname>:<port>/<path>/;auth=<authtype>;prot=<encryptiontype>
```

The <authtype> extension is optional, and should be "tls" if included; and <encryptiontype> can be either "p" (encrypted) or "c" (clear). If no <encryptiontype> extension is included, Fetch will assume the connection should be encrypted. An example FTPS URL would be:

```
ftps://ftp.example.com/www/;prot=c
```

When no port, or a port other than 990, is specified in an FTPS URL, Fetch will use the AUTH TLS method of connecting. You can specify SSL connect (implicit SSL) in a FTPS URL by using port 990:

```
ftps://arthur:@ftp.example.com:990/www/
```

Note that other file transfer clients may interpret FTPS URLs differently than Fetch.

Fetch also supports a special URL format for [Kerberos](#) FTP connections (**FTP with GSSAPI** and **FTP with KClient**):

```
x-ftp://<username>@<hostname>:<port>/<path>/;auth=<authtype>;prot=<encryptiontype>
```

Where <authtype> can be either "gssapi" or "kerberos\_v4", and <encryptiontype> can be either "p" (encrypted) or "c" (clear). If no <encryptiontype> extension is included, Fetch will assume the connection should be encrypted. An example x-ftp URL would be:

```
x-ftp://arthur:@kerberos.example.com/www/;auth=gssapi;prot=p
```

HTTP URLs, or web addresses, usually start with "http://" or "https://" and usually do not include usernames. For example:

```
http://www.example.com/www/photos.html
```

While Fetch cannot understand all web addresses or display all web content, when WebView is set up for a server, you can use the **Copy Web Address** command to copy a web address for selected items. You can also paste web addresses for that server in the New Connection dialog or the New Shortcut dialog to fill out the connection information, or in shortcut list windows to create shortcuts.



## Overview

When you put or upload a Macintosh file to a server, Fetch needs to know which format to use to store the file on the server. The best format to use depends on whether a file contains special Macintosh information, whether the file is a text or binary file, and whether you want to store the file or files in a compressed archive format.

Fetch's standard upload formats are **Automatic**, **Text**, **Binary (Raw Data)**, **MacBinary**, **BinHex**, and **AppleSingle**; there are also four special archive and compression formats: **Gzip**, **Stuffit X Archive**, **Zip Archive**, and **Gzipped Tar Archive**.

Normally Fetch uses the **Automatic** upload format, which examines each file and chooses the best format for it. However, if the **Automatic** upload format picks the wrong format for your files, you may want to choose one of the other formats at least temporarily. See below for more information about each format.

You can choose a different upload format from the [Format](#) submenu in the [Remote menu](#). Your new choice will apply only to the active [transfer window](#) until you close it. Different transfer windows may have different format settings. To set the initial upload format for new transfer windows, use the **Default upload format** preference in the [Upload Preferences](#) pane. You cannot set the archive and compression formats as the default upload format.

The [Put](#) files and folders prompt also has a **Format** menu. The format you choose in the **Put** prompt only applies for that one upload, and does not change the setting of any transfer windows.

If you change the upload format frequently, you can add a **Format** button to the [transfer window toolbar](#).

## Deciding on a format

The main considerations when uploading a file are whether the file contains special Macintosh information, and whether it is a text or binary file.

Files on the Macintosh either consist of files that have only a data fork (such as HTML files, JPEG files, and Microsoft Word files), or files that have both data and special information for the Macintosh in a resource fork as well.

Non-Macintosh computers generally only recognize the data fork part of a file, and standard methods for uploading and storing files on other computers only send the data fork. For cross-platform files, such as HTML and JPEG files, this is not a problem, since they are all data.

Care has to be taken when uploading files containing special Macintosh information to make sure those files will work properly when downloaded to another Macintosh. They must be encoded in a format that will preserve the Macintosh information by putting it all in the data fork. When you do this, however, the files will generally not be readable or usable on other computers. The upload formats that preserve Macintosh information are **MacBinary**, **BinHex**,

## AppleSingle, Stuffit X Archive, and Zip Archive.

If a file does not have special Macintosh information, the only decision to make is if it's a text file or a binary file. Text files contain lines of letters, numbers and punctuation; binary files may contain data in any format. Text files should be uploaded using the **Text** upload format, and plain binary files should be uploaded using the **Binary (Raw Data)** format.

Instead of making all these decisions yourself, the Fetch **Automatic** upload format examines each file and chooses the best of the upload formats for it. For more information about how the **Automatic** format makes its decisions, see the [Automatic upload format](#) help topic.

## Archive and compression formats

However, if you have a set of files and want to package them up as a single file for other users to download, instead of asking them to download all the files separately, you might want to consider one of the archive formats. When you upload one or more files using an archive format, all the files are combined into a single file. (The **Automatic** upload format will never combine multiple files into one file.) The archives are also compressed, so you may save space on the server by putting your files in an archive. Archive upload formats supported by Fetch are **Stuffit X Archive** (a compressed archive format common on the Macintosh), **Zip Archive** (a compressed archive format used by the Finder's Compress/Create Archive command, and also common on Windows), and **Gzipped Tar Archive** (a compressed archive format common in Unix). Even if other users are not using Fetch to download files you upload, the Stuffit and Zip formats are standard and most computers have tools to decode them (such as Stuffit Expander).

Fetch also supports uploading files in **Gzip** format, but Gzip is only a compression format, not an archive format — when you upload multiple files with Gzip, you end up with multiple compressed files.

Stuffit support is built into Fetch, so you do not need to have Stuffit installed on your Macintosh to create or decode files in the archive and compression formats.

## Format descriptions

**BinHex** is a format for representing a Macintosh file in text form. Because BinHex files are simply text, they can be sent through most electronic mail systems and stored on most computers. However, the conversion to text makes the file larger, so it takes longer to transmit a file in BinHex format than if the file was encoded some other way. For more information, see the [BinHex](#) help topic.

**MacBinary** is a common format for representing all the information in a Macintosh file (both data and resource forks) as a binary file. For more information, see the [MacBinary](#) help topic.

**AppleSingle** is a format created by Apple for representing a Macintosh file (both data and resource forks) on other types of computers in a binary file. For more information, see the [AppleSingle](#) help topic.

**Binary (Raw Data)** format uploads the data fork of a Macintosh file without any interpretation or conversion. It is a good choice for binary files that contain no special Macintosh information and that can be understood by different types of

computers, such as GIFs, JPEGs, Microsoft Word, and Excel files. For more information, see the [Binary \(Raw Data\)](#) format help topic.

**Text** format should only be used for files that consist only of letters, numbers, and punctuation, such as HTML files and files created by text editors like BBEdit. Line ending conversion is performed on files uploaded in Text format when uploaded using FTP, but not when using SFTP. If you have a text file whose name ends in an extension usually associated with binary files (such as ending in "bin", "Z", "sit", or "gif"), you'll need to choose the Text upload format to avoid the behavior of Automatic upload format. For more information see, the [Text format](#) help topic.

**Gzip** (or GNU zip) format is an open-source compression format common on Unix systems. Gzip only encodes the data fork of Macintosh files, so you should only use it for files that contain no special Macintosh information. If you have large files that will compress well and your target audience is Unix users, Gzip may be a good choice for them. Fetch's automatic decoding or StuffIt Expander can decompress Gzip files on the Macintosh. For more information, see the [Gzip](#) format help topic.

**StuffIt X Archive** format is a compression and archive format created by [Allume Systems](#) (formerly Aladdin Systems) as part of their StuffIt line of products, and is a common method for storing files on the Macintosh. StuffIt X Archive format preserves both data and resource forks of Macintosh files. If you want to combine a set of files together in one archive for downloading by other Macintosh users, StuffIt X Archive is a good choice. StuffIt Expander is also available for free for Windows. For more information, see the [StuffIt X Archive](#) format help topic.

**Zip Archive** format is a compression and archive format that is used by the Finder's Compress (Create Archive) command, and is also widely used on the Windows operating system. Like Compress (Create Archive) in the Finder, Fetch's Zip archives preserve both the data and resource forks of Macintosh files. If you want to combine a set of files together in one archive for downloading by Windows users, Zip Archive is a good choice. Fetch's automatic decoding or StuffIt Expander can decode Zip archives on the Macintosh. For more information, see the [Zip Archive](#) format help topic.

**Gzipped Tar Archive** format is a compression and archive format common on Unix systems. It uses Tar, a Unix archiving utility, to combine files into an archive, and then Gzip to compress the archive to make it smaller. Gzipped Tar archives only encode the data fork of Macintosh files, so you should only use it for files that contain no special Macintosh information. If you want to combine a set of files together in one archive for downloading by Unix users, Gzipped Tar Archive is a good choice. Fetch's automatic decoding or StuffIt Expander can decode Gzipped Tar archives on the Macintosh. For more information, see the [Gzipped Tar Archive](#) format help topic.

WebView lets you set up a correspondence between files you are viewing in Fetch and the address of the webpage they are part of, so that you can easily review changes to your site in a web browser and copy the web address of those files.

Once you've set up WebView for a server, you can then click the **WebView** button in the toolbar of transfer windows to show the selected files or folders in your web browser. You can also use the [Copy Web Address](#) command to copy a web address (also called a link or URL) of the selected items in Fetch.

This topic covers the concepts and features of WebView in depth; if you just want to get started, see the [setting up WebView](#) help topic.

## Fetch Addresses and Web Addresses

The address (URL) you use to access files in Fetch is not the same as the address you would use to access files in a web browser. For instance, a Fetch address might look like this:

```
ftp://arthur:@ftp.example.com/www/photos.html
```

whereas the corresponding web address would like this:

```
http://www.example.com/arthur/photos.html
```

Both of these addresses refer to the file "photos.html" that belongs to a user named "arthur", that is being hosted on the server "example.com". However, the first is an address that Fetch and other FTP clients understand, whereas the second is what someone would type into a web browser to see arthur's page of photos. (These types of addresses are also known as URLs; see the [Uniform Resource Locators \(URLs\)](#) topic for more information about them.)

If you have your own domain name, the addresses may look even more different; you may have something like:

```
Fetch address: ftp://arthur:@ftp.example.com/www/photos.html
```

```
Web address: http://www.WorldOfArthur.com/photos.html
```

Sending a Fetch address to most other people often isn't useful (unless they are also working with FTP), because even though most web browsers understand Fetch addresses, they won't show the file as a webpage when you use a Fetch address — instead may download the file or open an FTP client to view the file. Fetch addresses are also usually personalized, requiring your username and password to see the file, whereas web addresses are usually generalized and can be used by anyone without entering a password. So, once you have set up a WebView entry for your server, you can use the [Copy Web Address](#) command to get a web address, and send it to other people and they can click on it to view the webpage or file in a web browser.

## Configuring WebView

Before you can use WebView with a server, you must set up a correspondence between a folder in Fetch and the address of a webpage that folder represents.

Why doesn't Fetch automatically figure out the web address for you? Unfortunately, there's no standard for specifying the web address of files on a server; each server uses its own method, and the method used by one server may not apply on another. There is no way for Fetch to ask your server which method it uses, so Fetch has to ask you instead. While the information for some popular servers is built into Fetch, in most cases Fetch needs your input. If you are not sure what the corresponding web address should be, you should contact your hosting provider or server administrator.

In general, you want to create a WebView entry for your home folder on a server, or the folder that contains your webpages; then that entry will apply for every file and subfolder in that folder. If you create a WebView entry for a subfolder, the WebView entry may only apply for all the items in that subfolder, but not for the items in the subfolder's parent folder.

However, Fetch tries to generalize WebView entries if it can — if it notices that the same folder name is at the end of both the Fetch address and web address, it will remove that folder name from both addresses to create a WebView entry that will apply to other folders on the server as well. While this should almost always be what you want, if for some reason you need to have the same folder name at the end of each address, you can edit the WebView entry using the WebView Preferences pane, and Fetch will not remove matching folder names.

To set up WebView for a server:

1. Connect to your server using the [New Connection dialog](#) or by opening the shortcut for that server.
2. Go to your home folder, or the folder containing your webpages, if you are not already there.
3. Click the **WebView** button in the toolbar, or choose **Remote > WebView**.
4. The WebView dialog should appear. (If it does not, and a webpage opens instead, WebView is already set up for this server — either you set it up already, or you are on a server whose information is built into Fetch already. In either case, you're done.)
5. Look at the server and folder name mentioned in the dialog, or displayed in the window's title bar. You are setting up WebView for that folder.
6. Enter the web address corresponding to this folder and server in the edit field.
  - a. Usually you should enter your domain name, or the address for your website's homepage.
  - b. Examples of web addresses are:  
`http://www.example.com/myusername/` or  
`http://www.MyDomainName.com`
  - c. You can use the **Get From Safari** pop-up menu to get the web address of a webpage currently open in Safari.
7. Click the **OK** button. Fetch should show the selected file or folder in your web browser.

If the webpage that opens after you have entered a web address is incorrect — that is, it doesn't show you the page you expected, or it shows you an error — you may have entered an incorrect web address and need to try again. (See "Modifying WebView Entries" below.)

Once you have set up WebView for a server, you do not have to go through the set up process again.

The next time you choose **WebView** when connected to the same server, Fetch will show you the selected files or folder in your web browser instead of the WebView dialog.

Another way to set up WebView for a server is to choose **Fetch > Preferences**, click the [WebView](#) tab, and click the Add (+) button to add a new WebView entry using the expanded [WebView dialog](#).

When you use the Preferences to set up WebView, the WebView dialog will request two pieces of information, both the Fetch address and a web address. You need to either enter the Fetch address of the server and folder you want to create a WebView entry for, or you can use the **Get From Fetch** pop-up menu to the right of the Fetch address field to get the Fetch address of a currently open transfer window. Then enter the corresponding web address or use the **Get from Safari** popup menu to get the web address of a webpage currently open in Safari.

The **Get From Fetch** pop-up menu is only available when there are open [transfer windows](#), and the **Get From Safari** pop-up menu is only available when Safari is running and has open webpages. While WebView works with any web browser, Fetch can only show you a list of open webpages from Safari due to limitations of other browsers.

If you work with multiple servers, you can set up WebView entries for each server.

### Using WebView: WebView Command

The WebView command allows you to select items in Fetch, and view them in your web browser.

To do this, select an item in a Fetch file list, and click the **WebView** button in the toolbar or choose **Remote > WebView**. If you have already set up WebView — or if you are using a hosting provider whose information is built into Fetch — a page corresponding to that item will open in your web browser. If you selected an HTML file representing a webpage, you will see the webpage, or if you selected an image file, you'll see the image. If you select a folder, what you see in the web browser depends on how your server is configured; you may see the contents of the folder, you may see a webpage that's inside that folder (if that folder has its own index file), or you may get an error, because web browsers aren't allowed to see the contents of that folder directly.

If you select multiple files, Fetch will open multiple webpages.

If you have not already set up WebView for this folder or one of its parent folders, the WebView dialog will appear, prompting you to set up the WebView for this folder. See the "Configuring WebView" section above for how to do this.

### Using WebView: Copy Web Address

Fetch lets you copy the web address of the files (instead of the less useful Fetch address) by selecting files and choosing **Edit > Copy Web Address**. The web address is put on the clipboard, and you can use the Paste command to paste it into an email, a document you're editing, or in a web browser.

Once you've set up WebView for a server, you can also paste web addresses for that server in the New Connection dialog or the New Shortcut dialog to fill out the connection information, or in shortcut list windows to create shortcuts.

WebView is compatible with every major Mac OS X browser. It will use the browser you have specified as your default web browser. The WebView button's icon will always be the icon of your default web browser.

## Modifying WebView Entries

If you want to delete or edit a WebView entry you created earlier, you can use the [WebView Preferences](#) pane. Choose **Fetch > Preferences**, and click on the **WebView** tab. Then select an entry, and click the Delete (-) button to delete it, or click the Edit (✎) button to edit it. You can also add new entries by clicking the Add (+) button.

Alternately, you can edit a WebView entry from a Fetch transfer window. To do this, hold down the Option key and click the **WebView** button, or choose **Remote > WebView** while holding down the Option key. The WebView dialog will appear, and you can edit the web address entry that matches the folder or server displayed in that transfer window.

Related topics:

- [WebView dialog \(web address only\)](#)
- [WebView dialog \(with Fetch address\)](#)
- [WebView Preferences](#)
- [WebView command](#)
- [Copy Web Address](#)

The Zip Archive format is a compression and archive format that is used by the Finder's Compress (Create Archive) command, and is also widely used on the Windows operating system. The Zip Archive format takes multiple files and encodes them into a single file, compressing the data to take up less space at the same time. Zip archives typically have names ending with the extension ".zip". Fetch can create Zip archives when putting files, and automatically decode them when getting files.

Like Compress (Create Archive) in the Finder, Fetch's Zip archives preserve both data and resource forks, long file names, and other Macintosh Finder information. Zip Archive format is a good choice if you want to combine a set of files together in one archive for downloading by Windows users, or if you know that the users who will download your files use Mac OS X 10.3 or later (Finder archives were introduced in Mac OS X 10.3).

Fetch's automatic decoding can decode Zip archives on the Macintosh. StuffIt Expander can also decode Zip archives created by Fetch, however, StuffIt versions earlier than 9.0 may not properly decode the special Macintosh information encoded in the .zip files.

If a Windows user downloads a Zip archive created by Fetch, it may contain extra data that cannot be interpreted on Windows. Windows users can easily ignore this extra Macintosh-only information. There is no way to create a Zip archive without the extra Macintosh information when uploading in Fetch, although the extra information is usually only added when necessary.

See the [upload formats](#) help topic for information about encoding files in Zip Archive format when uploading. You cannot set Zip Archive format as your default upload format.

Fetch automatically recognizes and decodes Zip archives when downloading, although automatic recognition can be turned on and off by either:

- Selecting a remote file whose name ends with ".zip", choosing **Remote > Get Info** to show the [info window](#), and, in the [Transfer Options](#) pane, checking or unchecking the **Automatically decode files like this** checkbox — this will enable or disable decoding of Zip archives but will not affect the decoding of other formats, such as StuffIt archives;
- Or choosing **Fetch > Preferences**, clicking the [Download](#) tab, and checking or unchecking the **Allow automatic decoding of downloaded files** checkbox — this will enable or disable decoding of all file types.

For more information about using Zip archives in Fetch, see the [StuffIt and Archive formats](#) help topic.

Related topics:

- [StuffIt and Archive formats](#)
- [Upload formats](#)
- [Transfer Options](#)

 [Fetch Help](#) > [Reference](#) > **Messages**

The following topics offer information about some of the messages and warnings Fetch and might show you. Not all messages and warnings are covered. If you don't see what you're looking for, use the Search field above.

[Delete stray items at destination \(Mirror window\)](#)

[Host Authenticity](#)

["RSA host key differs" warning](#)

["The data received from the server was not secure" warning](#)

["Your password will be sent in the clear" warning](#)

["Your password will be sent in the clear to a secure server" warning](#)



## [Fetch Help](#) > [Messages](#) > Delete stray items at destination (Mirror window)

The **Delete stray items at destination** checkbox in the [Mirror window](#) instructs Fetch to delete items in the destination folder that do not exist in the source folder when mirroring two folders. This lets you synchronize the two folders so they are completely identical; normally mirroring just adds items to the destination folder or updates existing items, it does not remove items altogether.

**This can be a dangerous option, because it can delete entire folders and their contents.** If you accidentally select the wrong destination folder and you select **Delete stray items**, you could lose a lot of files. Be very sure you really want to delete non-matching items from the destination when you select this option, and use it with care!

A warning is displayed when you select this option to remind you of the possible dangers.

Related topics:

- [Mirror window](#)



## [Fetch Help](#) > [Messages](#) > Host Authenticity ("The authenticity of host can't be established" warning)

When Fetch warns you that "The authenticity of host 'ftp.example.com' can't be established, are you sure you want to continue connecting?", all it's really saying is that you're trying to connect using SFTP to a server that you have not connected to previously.

Each SFTP server has an SSH host key that enables secure transfers. When you connect to an SFTP server, Mac OS X stores that server's key for future use. When that happens, you have accepted the host key, saying that you trust the server.

Normally Fetch doesn't notify you when it accepts SSH host keys for servers you have not previously connected to, because accepting a key isn't generally regarded as a security risk. Also, you must accept the key if you want to transfer files from or to that server! However, if you want to confirm that you trust a server before connecting to it, you can check the **SFTP: Ask before accepting unknown host keys** preference in the [Security Preferences](#) pane. When that preference is checked, you will see the "The authenticity of host 'ftp.example.com' can't be established, are you sure you want to continue connecting?" warnings.

If there is a more serious problem with the SFTP connection — for instance, the SSH host key you have stored for a server does not match the key the server is now using, or the IP address of a server has changed but the key has not — Fetch will refuse the connection to the server. If you are having trouble connecting to a server that used to work, check to make sure you've entered the correct information in the New Connection dialog, and then contact the server administrator.

Related topics:

- [SFTP](#)
- [Security Preferences](#)

## [Fetch Help](#) > [Messages](#) > "RSA host key differs" warning

If Fetch displays a message similar to "Warning: the RSA host key for 'ftp.example.com' differs from the key for the IP address '192.0.34.166'" when you are opening a connection to an SFTP server, there may be a security problem with the server you are trying to connect to. The warning may also indicate that the host has legitimately changed its host key.

Each SFTP server has an SSH host key that enables secure transfers. When you connect to an SFTP server, Mac OS X stores that server's key for future use. This warning occurs when two different host keys for the same server have been stored in your `known_hosts` key file — one under the server's hostname, and one under the server's IP address.

While your server probably changed keys legitimately, this warning could indicate a security problem, and you should check with your server administrator or computing helpdesk before continuing with your connection. Click the **Cancel** button to stop connecting, or click **Connect** to go ahead and connect.

You cannot disable this warning in Fetch. If you repeatedly receive this warning when connecting to a server even after your server administrator or computing helpdesk verifies that there isn't a problem (or fixes the problem), the best way to stop receiving the warning is to delete your `~/ .ssh/known_hosts` file.

Important: Consult your server administrator or computing helpdesk for assistance before deleting this file if you are not comfortable with these instructions.

To delete the `known_hosts` file, click on this link:

[Delete the known\\_hosts file for me](#)

Or you can follow these steps:

1. In the Finder, choose **Go > Go to Folder**. The Finder will display a prompt.
2. Type "`~/ .ssh`" (without the quotes) into the prompt, and click the **Go** button. The Finder will display a window showing the contents of the `.ssh` folder.
3. Select the `known_hosts` file, and drag it to the Trash.
4. Close the window.

The next time you try to connect to the server, you should not get a warning.

After deleting the `known_hosts` file, the first time you connect to a server you may be warned that the authenticity of the host cannot be established. This is normal if you have checked the **SFTP: Ask before accepting unknown host keys** preference in the [Security Preferences](#) pane. See the [Host Authenticity](#) topic for more information.

Related topics:

- [Host Authenticity](#)
- [SFTP](#)



[Fetch Help](#) > [Messages](#) > "The data received from the server was not secure"

## warning

Fetch displays the warning, "The data received from the server was not secure. This could mean that the server, or your connection to it, has been compromised," when you have a secure, encrypted Kerberos connection to a server and the server starts sending back unencrypted data.

If this happens, it could mean that the server, or your connection to it, has been compromised. You should disconnect and consult your network administrator or service provider.

Click the **Ignore** button to continue connecting, or **Disconnect** to not make the connection.

This warning should never appear when using SFTP or FTP with TLS/SSL to connect to a server.

Consult your network administrator or service provider if you have any question about whether the warning is significant.

You can turn off this warning by unchecking the **Always check protection level on incoming data** checkbox in the [Security Preferences](#) pane.

Related topics:

- [Security Preferences](#)



## [Fetch Help](#) > [Messages](#) > "Your password will be sent in the clear" warning

Fetch displays the warning, "If you do not disconnect, your password will be sent over the network in the clear," whenever you try to send your password unencrypted and insecurely.

There are two reasons you could be receiving this warning.

First, you will always receive this warning when connecting to a server that doesn't support secure connections. Most FTP servers do not support secure connections and require unencrypted passwords, so this warning does not have much value when connecting to them. You may want to contact your service provider or network administrator to see if your server supports secure connections using SFTP or FTP with TLS/SSL; if not, you may want to turn off this warning.

However, if you are connecting to a secure server via SFTP, FTP with TLS/SSL, Kerberos (GSSAPI or KClient), or One-Time Password, this warning may indicate that the secure server isn't working properly and requested a password insecurely, in which case, you may not want to continue connecting. Also, if you usually connect to secure servers, this warning may also indicate you accidentally tried to connect to an insecure server.

Click the **Ignore** button to continue connecting, or **Disconnect** to not make the connection.

Consult your service provider or network administrator if you have any question about whether the warning is significant.

You can turn off this warning by unchecking the **Warn before sending password insecurely** checkbox in the [Security Preferences](#) pane.

Related topics:

- [Security Preferences](#)
- [Security](#)

 [Fetch Help](#) > [Messages](#) > "Your password will be sent in the clear to a secure server" warning

Fetch displays the warning, "If you do not disconnect, your password will be sent over the network in the clear to a server that supports security," whenever you try to send your password unencrypted and insecurely to a server that says it supports a more secure way to send passwords; in other words, you are not taking advantage of the server's security support.

There are two reasons you could be receiving this warning.

Usually you will receive this warning when connecting to a server that supports Kerberos, and you didn't specify **FTP with KClient** or **FTP with GSSAPI** as the connection type in the New Connection dialog when connecting to the server. You could also receive it even when you did select one of those options because the server is not working properly; in that case, you should contact your network administrator. You should never see this warning when connecting to a server via SFTP.

However, you may also see this warning when connecting to a server that supports TLS/SSL, or a security system that is not recognized by Fetch. To avoid getting a warning in this case, either connect to the server using SFTP (if available), or uncheck the **Warn before sending password insecurely to Kerberos servers** preference in the [Security Preferences](#) pane.

Click the **Ignore** button to continue connecting, or **Disconnect** to not make the connection.

Consult your network administrator or service provider if you have any question about whether the warning is significant.

You can turn off this warning by unchecking the **Warn before sending password insecurely to Kerberos servers** checkbox in the [Security Preferences](#) pane.

There is also a more [general warning](#) that applies whenever you try to send your password in the clear, regardless of whether a server supports security or not.

Related topics:

- [Security Preferences](#)
- [Kerberos](#)
- [Security](#)



This is a short list of terms commonly used in Fetch that you should be familiar with. For a comprehensive list of all terms that you may not be familiar with, see the [Glossary](#) help topic.

**account** – When you have permission to access an FTP or SFTP server by entering a username and password, you are said to have an account on that server. A *home folder* to store your files is often associated with an account.

**anonymous FTP** – The process of connecting to a server that does not require any identification or password. The username used is either blank or "anonymous" and the password is optional. When you are connected to an anonymous FTP server, you are said to be a *guest* of the server. Several repositories of Macintosh software are available to all by anonymous FTP.

**binary file** – A file that may contain any data in any format; not a text file. Files in [MacBinary](#) format or [AppleSingle](#) format are binary files, as are GIF and JPEG image files, MP3 and WAV audio files, and QuickTime and MPEG video files.

**client** – FTP and SFTP, like many other Internet services, are divided into two parts — the *server* computer, which hosts the files and makes them available on the Internet, and the *client* computer, which accesses the files on or sends files to the server. Fetch is an FTP and SFTP client — it can read or send files to servers, but it cannot make files residing on your Macintosh available to everyone. When you use Fetch, your Macintosh is acting as the client computer.

**connection** – When you start to exchange data over the network with another computer, you have established a connection to that server. In Fetch, when a transfer window opens, you have opened a connection. The connection stays open while you look at files on the server or transfer files. When you close a *transfer window* in Fetch, you have closed your connection — you are no longer communicating with the server in any way. Some servers may automatically close a connection if it has been inactive for some time.

**connection type** – The method used to connect to a server. Fetch supports regular FTP connections, and secure connections using SFTP, FTP with TLS/SSL, and Kerberos (GSSAPI and KClient). See the [connection types](#) help topic for more information.

**current folder** – The remote folder you are currently viewing in a *transfer window*. Most Fetch actions, such as putting files or creating new folders, will take place in the current folder unless otherwise specified. Sometimes referred to as the "working folder" or "working directory."

**directory** – A collection of files and folders, equivalent to Macintosh folders. While some operating systems (such as UNIX) refer to folders as directories, in Fetch, directories are always referred to as *folders* or *remote folders*.

**download** – The process of moving a file from some other computer to your own. In Fetch, also referred to as *get*. When you are getting a file with Fetch, you are downloading it.

**download folder** – The folder on your Macintosh where Fetch will put your files when you double-click a remote file to download it, or use the **Get** command. Specified in the [Download Preferences](#) pane.

**download mode** – When Fetch gets a file, it must request the file as either a text file or as a

binary file — that is, the file is to be downloaded in either text mode or binary mode. If a binary file (such as a JPEG file) is downloaded in text mode, it will become corrupted and unusable, so it is important to use the right mode. Fetch has an Automatic download mode that works well for most situations so that you don't have to decide each time. See the [download modes](#) help topic for further information.

**encryption** – The act of scrambling data so that only the receiver can read it, protecting the privacy of the data while it is being transferred over the Internet. SFTP connections always encrypt your data, while FTP with TLS/SSL or Kerberos connections can optionally encrypt the data.

**extension** – A suffix after a period at the end of a filename that identifies what kind of file it is. For example, the extension of "readme.txt" is ".txt", indicating it is a text file; the extension in "dog.jpg" is ".jpg", indicating a JPEG file, and so on. Some files on your Macintosh or on servers may not have extensions, in which case you may have to guess what kind of file they are.

**Fetch address** – An address that refers to files on an FTP or SFTP server in Fetch. A Fetch address is a kind of *URL*. An example of a Fetch address is "ftp://arthur:@ftp.example.com/www/photos.html". You can use Fetch addresses to tell Fetch which file or folder to access, and Fetch addresses can be associated with *web addresses* using WebView. See the [WebView](#) and [URLs](#) help topics for more information.

**file list** – The main part of the *transfer window* showing the files and folders of the *current folder*. It looks and works much like the Mac OS Finder's list view.

**folder** – A collection of files and subfolders. Referred to as a *directory* on some systems. Subfolders are folders within a folder; the enclosing folder is known as the parent folder. Most systems have a notion of the *current folder*, the folder whose files you are viewing at that time. Not all systems have folders, and some systems do not permit subfolders.

**FTP** – FTP stands for "File Transfer Protocol", and is a set of ground rules (a "protocol") that allows two computers to exchange files over a network. Fetch is an FTP client program. See the [FTP](#) help topic for more information.

**FTP with TLS/SSL** – A set of ground rules (a "protocol") that allows two computers to exchange files securely over a network. When you connect to a server using FTP with TLS/SSL, encryption is used to protect your password and optionally the data transferred between your Macintosh and the server. FTP with TLS/SSL uses the same protocol as FTP, but is wrapped in an encryption layer, and should not be confused with SFTP, which is a completely separate protocol from FTP. Fetch supports two kinds of FTP with TLS/SSL: AUTH TLS, also known as FTPES or Explicit SSL; and SSL connect, also known as Implicit SSL. For more information, see the [FTP with TLS/SSL](#) help topic.

**FTPS** – A common synonym for [FTP with TLS/SSL](#). Not to be confused with SFTP.

**get** – The process of moving a file from some other computer to your own. Also referred to as *download*. When you are getting a file with Fetch, you are downloading it.

**guest** – A user of an *anonymous FTP* server. When you are connected to an anonymous FTP server, you are a guest of that server.

**home folder** – Your default folder on a server. If you do not specify an initial folder or path for a new connection in Fetch, a server will take you to the home folder for your username. For instance, on a web server, the home folder is often the folder where your webpages are

stored. The home folder is usually not the same as the *root folder* on the server, but you may not have permission to see the parent or any enclosing folders of the home folder. Sometimes referred to as the *remote home folder*.

**host** – The remote computer you are connected to. Also referred to as the *server* — see the *server* entry below.

**hosting service, hosting company** – The company, organization, or institution that maintains a particular server that you use and grants you access to the server. Compare with *Internet service provider*, which is the company that provides you access to the Internet in general. While many Internet service providers also provide servers (such as web servers for your webpages), you may also access servers that are not maintained by your Internet service provider. For instance, you may use your cable Internet connection (the cable company is your Internet service provider) to transfer files to a server at your office (your company is the hosting service for that server). Fetch Softworks is not a hosting company or Internet service provider.

**hostname** – The name used to identify a computer on the Internet, for example "ftp.fetchsoftworks.com". This is usually how you refer to a server when connecting to it with Fetch. Hostnames form the first part of most *URLs*.

**Internet** – A collection of networks linking schools, universities, companies, and individuals around the world. FTP and SFTP are standard file transfer protocols of the Internet.

**Internet provider, Internet service provider (ISP)** – The company, organization, or institution that provides you with access to the Internet. Compare with *hosting service*, which is the company that maintains and gives you access to a particular server. See the *hosting service* entry above for more information. Fetch Softworks is not an Internet service provider or hosting service.

**media file** – An image, sound, movie, animation, PDF, or any other sort of file that can be displayed by Apple's QuickTime software.

**parent, parent folder** – The folder that encloses another folder. If folder "Pictures" is inside folder "My Files", "My Files" is the parent of "Pictures".

**path** – A description of a file or folder's location that includes the folders the item is in, for example, "/www/pictures/dog.jpg" is the path to the file "dog.jpg" that is in the folder "pictures" that in turn is in the folder "www". Differs from a *URL* in that it does not completely specify an item's location on the Internet, only on the current server. A path can be either a full path (one that is specified in relation to the root folder) or a relative path (one that is specified in relation to the current folder). A path is sometimes called a pathname.

**put** – The process of moving a file from your computer to another one. Also referred to as *upload*. When you put a file with Fetch, you are uploading it.

**root folder** – The topmost folder on a server; the folder that is the *parent* of all folders on a server. Usually represented in folder listings as "/" (a slash). The root folder may be different from the topmost folder you have access to — for example, on a web server, you may not have access to the parent folders of your website's folder, even though there are folders enclosing it. The root folder is sometimes just called "root" or even "/" (slash).

**server** – The remote computer you are connected to and transferring files from or to. The server runs a program that accepts your connections; your Macintosh is the *client*. Most servers are computers dedicated to hosting and serving files, but a personal computer can also

act as a server. Sometimes a server is referred to as the *host* .

**session** – The time spent and actions taken between opening a connection to a server and closing the connection is an FTP (or SFTP) session, or just session for short. Each transfer window represents a session.

**SFTP** – A set of ground rules (a "protocol") that allows two computers to exchange files securely over a network. When you connect to a server using SFTP, SSH encryption is used to protect your password and the data transferred between your Macintosh and the server. Fetch supports connecting to servers with SFTP. For more information, see the [SFTP](#) help topic. SFTP stands for "SSH File Transfer Protocol," although it is also sometimes referred to as "Secure File Transfer Protocol."

**shortcut** – A quick way to access a file or folder available over the Internet via FTP or SFTP, similar to bookmarks or favorites in web browsers. See the [shortcuts](#) help topic for more information.

**TCP/IP** – A protocol suite developed for the United States Department of Defense, and used by many types of computers (notably, all the computers on the Internet). FTP and SFTP, the protocols used by Fetch, are part of the TCP/IP family of protocols. Fetch uses TCP/IP services provided in Mac OS X. TCP/IP stands for "Transmission Control Protocol/Internet Protocol."

**text file** – A file containing lines of letters, numbers, and punctuation. Most word processors can create text files and some editors (such as BBEdit) only create text files. A non-text file is a *binary file*.

**transfer window** – The central window in Fetch for viewing or modifying files on a server, and for transferring files between your Macintosh and a server. A transfer window appears once you have successfully opened a connection to a server. For more information, see the [transfer window](#) help topic.

**URL (Uniform Resource Locator)** – A global name for a resource available on the Internet. An example of a URL is "ftp://ftp.fetchsoftworks.com/example/Fetch\_5.6.dmg". In this case, the resource is a file, available for FTP on host "ftp.fetchsoftworks.com", with the path "example/Fetch\_5.6.dmg". You can use URLs to tell Fetch which file or folder to access. URLs were developed as part of the World Wide Web (WWW) system on the Internet. See the [URLs](#) help topic for more information.

**upload** – The process of moving a file from your computer to another one. In Fetch, also referred to as *put*. When you put a file with Fetch, you are uploading it.

**upload format** – The format, or encoding, that Fetch will use when you put or upload a file to a server. Fetch must decide whether to store files as *text files* or *binary files*, and how to encode Macintosh files that contain special information in order to preserve that information (using the BinHex, MacBinary, AppleSingle, or Stuffit X formats). Fetch has an Automatic upload format which determines the best upload format to use for each file. See the [upload formats](#) help topic for more information.

**web address** – An address you use to refer to files or webpages in a web browser. When you type an address into a web browser, you are using a web address. An example of a web address is "http://www.example.com/arthur/photos.html". A web address is a kind of *URL*. Web addresses can be associated with *Fetch addresses* using WebView. See the [WebView](#) and [URLs](#) help topics for more information.



This is a comprehensive list of terms used in Fetch that you may not be familiar with. For a shorter list of the essential terms to understand, see the [Terms You Should Know](#) help topic.

**account** – When you have permission to access an FTP or SFTP server by entering a username and password, you are said to have an account on that server. A *home folder* to store your files is often associated with an account.

**active FTP** – One of two ways of establishing an FTP connection, in which Fetch tells an FTP server how to connect to your Macintosh. Compare with *passive FTP*, and see the [Active and Passive FTP](#) help topic for more information.

**anonymous FTP** – The process of connecting to a server that does not require any identification or password. The username used is either blank or "anonymous" and the password is optional. When you are connected to an anonymous FTP server, you are said to be a *guest* of the server. Several repositories of Macintosh software are available to all by anonymous FTP.

**AppleSingle** – A format for representing a Macintosh file as a single binary file in order to preserve special Macintosh information that cannot be stored properly on some systems. For more information, see the [AppleSingle](#) help topic.

**archive** – A kind of file in which multiple files and folders are encoded together into a single file. The archive can be decoded later back into the original files and folders. Archives often also compress the files, saving space.

**authentication** – A process that proves that you are who you claim to be. Entering a username and a password is an example of authenticating yourself. If you enter a username but cannot enter the password correctly, you have not authenticated yourself, because entering the correct password is the only way to prove that you are the user identified by that username.

**binary file** – A file that may contain any data in any format; not a text file. Files in [MacBinary](#) format or [AppleSingle](#) format are binary files, as are GIF and JPEG image files, MP3 and WAV audio files, and QuickTime and MPEG video files.

**BinHex** – A format for representing a Macintosh file in text form, in order to preserve special Macintosh information that cannot be stored properly on some systems. For more information, see the [BinHex](#) help topic.

**client** – FTP and SFTP, like many other Internet services, are divided into two parts — the *server* computer, which hosts the files and makes them available on the Internet, and the *client* computer, which accesses the files on or sends files to the server. Fetch is an FTP and SFTP client — it can read or send files to servers, but it cannot make files residing on your Macintosh available to everyone. When you use Fetch, your Macintosh is acting as the client computer.

**connection** – When you start to exchange data over the network with another computer, you have established a connection to that server. In Fetch, when a transfer window opens, you have opened a connection. The connection stays open while you look at files on the server or transfer files. When you close a *transfer window* in Fetch, you have closed your connection — you are no longer communicating with the server in any way. Some servers may automatically close a connection if it has been inactive for some time.

**connection type** – The method used to connect to a server. Fetch supports regular FTP connections, and secure connections using SFTP, FTP with TLS/SSL, and Kerberos (GSSAPI and KClient). See the [connection types](#) help topic for more information.

**current folder** – The remote folder you are currently viewing in a *transfer window*. Most Fetch actions, such as putting files or creating new folders, will take place in the current folder unless otherwise specified. Sometimes referred to as the "working folder" or "working directory."

**data fork** – Some files on the Macintosh store their information in two different "forks," a data fork and a resource fork. The data fork usually contains the main body of the file's information, such as the text or graphic data that make up the file. The *resource fork* contains special Macintosh information. Most cross-platform files, such as HTML files, JPEG files, and Microsoft Word files, have just a data fork. See the [upload formats](#) help topic for a discussion of forks and how they relate to transferring files.

**directory** – A collection of files and folders, equivalent to Macintosh folders. While some operating systems (such as UNIX) refer to folders as directories, in Fetch, directories are always referred to as *folders* or *remote folders*.

**dot file** – A file whose name begins with a period.

**download** – The process of moving a file from some other computer to your own. In Fetch, also referred to as *get*. When you are getting a file with Fetch, you are downloading it.

**download folder** – The folder on your Macintosh where Fetch will put your files when you double-click a remote file to download it, or use the **Get** command. Specified in the [Download Preferences](#) pane.

**download mode** – When Fetch gets a file, it must request the file as either a text file or as a binary file — that is, the file is to be downloaded in either text mode or binary mode. If a binary file (such as a JPEG file) is downloaded in text mode, it will become corrupted and unusable, so it is important to use the right mode. Fetch has an Automatic download mode that works well for most situations so that you don't have to decide each time. See the [download modes](#) help topic for further information.

**droplet** – A small application, usually with no or a minimal user interface, that you drag files to in order to perform a certain action. In Fetch, you can create [droplet shortcuts](#) which upload files to a specific server, or you can use [Automator](#) and Fetch's [Automator actions](#) to create more complicated droplets that upload files and then perform other actions on them afterwards.

**encoding** – Changing data into another format. For instance, substituting numbers for letters (A=1, B=2, etc.) is a simple type of encoding. Fetch can encode files so that they can be transferred in a certain way without damaging them, such as encoding a file in BinHex to prevent the special Macintosh information from being lost. Can also refer to character encoding, the character set a server is using to display item names or text files.

**encryption** – The act of scrambling data so that only the receiver can read it, protecting the privacy of the data while it is being transferred over the Internet. SFTP connections always encrypt your data, while FTP with TLS/SSL or Kerberos connections can optionally encrypt the data.

**extension** – A suffix after a period at the end of a filename that identifies what kind of file it is. For example, the extension of "readme.txt" is ".txt", indicating it is a text file; the extension

in "dog.jpg" is ".jpg", indicating a JPEG file, and so on. Some files on your Macintosh or on servers may not have extensions, in which case you may have to guess what kind of file they are.

**Fetch address** – An address that refers to files on an FTP or SFTP server in Fetch. A Fetch address is a kind of *URL*. An example of a Fetch address is "ftp://arthur:@ftp.example.com/www/photos.html". You can use Fetch addresses to tell Fetch which file or folder to access, and Fetch addresses can be associated with *web addresses* using *WebView*. See the [WebView](#) and [URLs](#) help topics for more information.

**file list** – The main part of the *transfer window* showing the files and folders of the *current folder*. It looks and works much like the Mac OS Finder's list view.

**firewall** – Hardware or software that attempts to protect computers by preventing computers outside the firewall from starting connections with computers inside the firewall. Generally, your local network is inside, or behind, the firewall, whereas the rest of the Internet is outside the firewall. Firewalls only allow computers inside the firewall to start connections with outside computers.

**folder** – A collection of files and subfolders. Referred to as a *directory* on some systems. Subfolders are folders within a folder; the enclosing folder is known as the parent folder. Most systems have a notion of the *current folder*, the folder whose files you are viewing at that time. Not all systems have folders, and some systems do not permit subfolders.

**format** – Either the way a file is encoded, or the way a file will be uploaded. See the *upload format* definition below and the [upload formats](#) help topic for more information.

**FTP** – FTP stands for "File Transfer Protocol," and is a set of ground rules (a "protocol") that allows two computers to exchange files over a network. Fetch is an FTP client program. See the [FTP](#) help topic for more information.

**FTP with TLS/SSL** – A set of ground rules (a "protocol") that allows two computers to exchange files securely over a network. When you connect to a server using FTP with TLS/SSL, encryption is used to protect your password and optionally the data transferred between your Macintosh and the server. FTP with TLS/SSL uses the same protocol as FTP, but is wrapped in an encryption layer, and should not be confused with SFTP, which is a completely separate protocol from FTP. Fetch supports two kinds of FTP with TLS/SSL: AUTH TLS, also known as FTPES or Explicit SSL; and SSL connect, also known as Implicit SSL. For more information, see the [FTP with TLS/SSL](#) help topic.

**FTPES** – A common synonym for Explicit SSL, a kind of [FTP with TLS/SSL](#).

**FTPS** – A common synonym for [FTP with TLS/SSL](#). Not to be confused with SFTP.

**FXP** – FXP stands for "File eXchange Protocol," and is a way to transfer files directly between two FTP servers without moving the files to your Macintosh (the client) first. In Fetch, FXP is referred to as "direct server-to-server transfers." See the [server-to-server transfers](#) help topic for more information.

**get** – The process of moving a file from some other computer to your own. Also referred to as *download*. When you are getting a file with Fetch, you are downloading it.

**GSSAPI** – Stands for "Generic Security Service API," a programming interface to Kerberos v5. Fetch can make secure connections by using FTP with GSSAPI. See the [Kerberos](#) help topic for more information.

**guest** – A user of an *anonymous FTP* server. When you are connected to an anonymous FTP server, you are a guest of that server.

**Gzip** – A compression format commonly used on Unix. A Gzipped Tar Archive is another common *archive* format on Unix.

**home folder** – Your default folder on a server. If you do not specify an initial folder or path for a new connection in Fetch, a server will take you to the home folder for your username. For instance, on a web server, the home folder is often the folder where your webpages are stored. The home folder is usually not the same as the *root folder* on the server, but you may not have permission to see the parent or any enclosing folders of the home folder. Sometimes referred to as the *remote home folder*.

**host** – The remote computer you are connected to. Also referred to as the *server* — see the *server* entry below.

**hosting service, hosting company** – The company, organization, or institution that maintains a particular server that you use and grants you access to the server. Compare with *Internet service provider*, which is the company that provides you access to the Internet in general. While many Internet service providers also provide servers (such as web servers for your webpages), you may also access servers that are not maintained by your Internet service provider. For instance, you may use your cable Internet connection (the cable company is your Internet service provider) to transfer files to a server at your office (your company is the hosting service for that server). Fetch Softworks is not a hosting company or Internet service provider.

**hostname** – The name used to identify a computer on the Internet, for example, "ftp.fetchsoftworks.com". This is usually how you refer to a server when connecting to it with Fetch. Hostnames form the first part of most *URLs*.

**Internet** – A collection of networks linking schools, universities, companies, and individuals around the world. FTP and SFTP are standard file transfer protocols of the Internet.

**Internet provider, Internet service provider (ISP)** – The company, organization, or institution that provides you with access to the Internet. Compare with *hosting service*, which is the company that maintains and gives you access to a particular server. See the *hosting service* entry above for more information. Fetch Softworks is not an Internet service provider or hosting service.

**IP address** – A type of computer address of the form number.number.number.number, such as 192.0.34.166 . Usually you will refer to servers by their hostnames (such as "ftp.fetchsoftworks.com") but Fetch can also connect to servers using just their numeric address.

**KClient** – A popular interface to Kerberos v4 on the Macintosh. Fetch can make secure connections by using FTP with KClient. See the [Kerberos](#) help topic for more information.

**Kerberos** – Kerberos is a network security system that allows two computers on an insecure network to trust each other, and provides ways for the computers to exchange information securely. Fetch supports secure connections to FTP servers using Kerberos. For more information, see the [Kerberos](#) help topic.

**MacBinary** – A format for representing a Macintosh file as a single binary file, in order to preserve special Macintosh information that cannot be stored properly on some systems. For more information, see the [MacBinary](#) help topic.

**media file** – An image, sound, movie, animation, PDF, or any other sort of file that can be displayed by Apple's QuickTime software.

**mirror** – The process of synchronizing files between two computers by transferring only the ones that have changed. Mirroring is an efficient way to upload a website, since only new and changed files are transferred.

**mode** – The way Fetch requests a file for downloading — either text mode or binary mode. Fetch also has an Automatic mode that decides which mode to use based on the file being downloaded. See the *download mode* entry above and the [download modes](#) help topic for further information.

**NAT** – Short for "Network Address Translation." Hardware or software that allows multiple computers to share a single IP address. Often, NATs are part of another piece of equipment, such as an AirPort Base Station or a router.

**parent, parent folder** – The folder that encloses another folder. If folder "Pictures" is inside folder "My Files", "My Files" is the parent of "Pictures".

**passive FTP** – One of two ways of establishing an FTP connection, in which Fetch opens a connection to the FTP server directly. Passive FTP is more compatible with firewalls, routers, and NAT devices than *active FTP*. See the [Active and Passive FTP](#) help topic for more information.

**path** – A description of a file or folder's location that includes the folders the item is in, for example, `"/www/pictures/dog.jpg"` is the path to the file "dog.jpg" that is in the folder "pictures" that in turn is in the folder "www". Differs from a *URL* in that it does not completely specify an item's location on the Internet, only on the current server. A path can be either a full path (one that is specified in relation to the root folder) or a relative path (one that is specified in relation to the current folder). A path is sometimes called a pathname.

**permissions** – Permissions restrict which users can read, write, or execute files, or which users can view, get, or put files in a folder. For instance, you may have permission to look at and download a file on a server (read permission), but not permission to delete or make changes (write permission). For more information, see the [permissions](#) help topic.

**Point-to-Point Protocol (PPP)** – A protocol for using network protocols, including TCP/IP, over a dial-up modem connection or other serial line. You can transfer files with Fetch after you've established a connection to an Internet service provider that supports PPP (almost all do) using the PPP software included with Mac OS X.

**port** – Computers assign a number to each *connection* they have with another computer in order to keep track of simultaneous connections. These are called port numbers, or ports for short. Most servers listen for requests for FTP or SFTP connections on a standard port, but sometimes servers offer or require the ability to connect to them on a different port. You can tell Fetch can make FTP or SFTP connections to non-standard ports, but most of the time you do not need to.

**preferred application** – the application a file will open in after you've downloaded it with Fetch. Also, the application used to determine the appropriate icon to display for a file in the file list.

**proxy server** – A special server that acts as an intermediary between a client and the server that the client really wants to connect to. Proxy servers are used on networks where most computers are not permitted to make Internet connections; instead those computers must

connect to the proxy, which in turn connects to the desired server. Fetch supports a number of proxy servers (see the [Proxy Preferences](#) topic). Web proxy servers, which use the HTTP protocol rather than FTP, are not supported.

**put** – The process of moving a file from your computer to another one. Also referred to as *upload*. When you put a file with Fetch, you are uploading it.

**resource fork** – Some files on the Macintosh store their information in two different "forks," a data fork and a resource fork. The resource fork contains special Macintosh information that may need to be preserved for the file to work on another Macintosh after being transferred. The *data fork* usually contains the main body of the file's information, such as the text or graphic data that make up a HTML or JPEG file. See the [upload formats](#) help topic for a discussion of forks and how they relate to transferring files.

**root folder** – The topmost folder on a server; the folder that is the *parent* of all folders on a server. Usually represented in folder listings as "/" (a slash). The root folder may be different from the topmost folder you have access to — for example, on a web server, you may not have access to the parent folders of your website's folder, even though there are folders enclosing it. The root folder is sometimes just called "root" or even "/" (slash).

**server** – The remote computer you are connected to and transferring files from or to. The server runs a program that accepts your connections; your Macintosh is the *client*. Most servers are computers dedicated to hosting and serving files, but a personal computer can also act as a server. Sometimes a server is referred to as the *host*.

**session** – The time spent and actions taken between opening a connection to a server and closing the connection is an FTP (or SFTP) session, or just session for short. Each transfer window represents a session.

**SFTP** – A set of ground rules (a "protocol") that allows two computers to exchange files securely over a network. When you connect to a server using SFTP, SSH encryption is used to protect your password and the data transferred between your Macintosh and the server. Fetch supports connecting to servers with SFTP. For more information, see the [SFTP](#) help topic. SFTP stands for "SSH File Transfer Protocol," although it is also sometimes referred to as "Secure File Transfer Protocol."

**shortcut** – A quick way to access a file or folder available over the Internet via FTP or SFTP, similar to bookmarks or favorites in web browsers. See the [shortcuts](#) help topic for more information.

**Stuffit** – A kind of file that is a compressed archive of multiple files, a Stuffit archive (or Stuffit X archive). Also refers to the set of applications produced by [Allume Systems](#) for creating and decoding Stuffit archives, such as Stuffit Expander or Stuffit Deluxe.

**subfolder** – A folder inside another folder. If the folder "Pictures" is inside the folder "My Files", "Pictures" is a subfolder of "My Files". On some systems, referred to as a subdirectory.

**symbolic link** – On a server, a pointer to a file or folder in a different place. Similar to a Macintosh alias file.

**TCP/IP** – A protocol suite developed for the United States Department of Defense, and used by many types of computers (notably, all the computers on the Internet). FTP and SFTP, the protocols used by Fetch, are part of the TCP/IP family of protocols. Fetch uses TCP/IP services provided in Mac OS X. TCP/IP stands for "Transmission Control Protocol/Internet Protocol."

**text file** – A file containing lines of letters, numbers, and punctuation. Most word processors

can create text files and some editors (such as BBEdit) only create text files. A non-text file is a *binary file*.

**transcript** – A complete, unfiltered record of Fetch's "conversations" with FTP (or SFTP) servers – all of the commands Fetch sends, and the servers' responses. The transcript contains mostly technical and extraneous information that usually you do not need to see. See the [Fetch Transcript](#) help topic for more information.

**transfer window** – The central window in Fetch for viewing or modifying files on a server, and for transferring files between your Macintosh and a server. A transfer window appears once you have successfully opened a connection to a server. For more information, see the [transfer window](#) help topic.

**URL (Uniform Resource Locator)** – A global name for a resource available on the Internet. An example of a URL is "ftp://ftp.fetchsoftworks.com/example/Fetch\_5.6.dmg". In this case, the resource is a file, available for FTP on host "ftp.fetchsoftworks.com", with the path "example/Fetch\_5.6.dmg". You can use URLs to tell Fetch which file or folder to access. URLs were developed as part of the World Wide Web (WWW) system on the Internet. See the [URLs](#) help topic for more information.

**upload** – The process of moving a file from your computer to another one. In Fetch, also referred to as *put*. When you put a file with Fetch, you are uploading it.

**upload format** – The format, or encoding, that Fetch will use when you put or upload a file to a server. Fetch must decide whether to store files as *text files* or *binary files*, and how to encode Macintosh files that contain special information in order to preserve that information (using the BinHex, MacBinary, AppleSingle, or Stuffit X formats). Fetch has an Automatic upload format which determines the best upload format to use for each file. See the [upload formats](#) help topic for more information.

**verbose file list** – An extended text-only file list for a folder that often includes information, such as file permissions, that Fetch does not display in its regular file list. However, it is in a less user-friendly format than and not interactive like the file list of the transfer window.

**web address** – An address you use to refer to files or webpages in a web browser. When you type an address into a web browser, you are using a web address. An example of a web address is "http://www.example.com/arthur/photos.html". A web address is a kind of *URL*. Web addresses can be associated with *Fetch addresses* using *WebView*. See the [WebView](#) and [URLs](#) help topics for more information.

**Zip archive** – A compressed *archive* format used by the Finder's Compress (Create Archive) command, and also commonly used on computers running the Windows operating system.