

## About Fetch...

Fetch is a Macintosh program for transferring files using the File Transfer Protocol (FTP) on a TCP/IP network, such as the Internet. This allows a networked Macintosh to exchange files with any FTP server on the same network. Fetch runs on Mac OS System 7.0 or newer.

If you have questions or comments about Fetch, please visit the message board at <http://fetchsoftworks.com>.

This copy of Fetch is version 4.0.3

Fetch is Copyright © 2000, Trustees of Dartmouth College, © 2002, Fetch Softworks. This copy of Fetch is sold under license by Fetch Softworks. Other product and company names mentioned herein may be trademarks or trade names of their respective holders.

Fetch is licensed free of charge to users closely affiliated with educational or non-profit charitable organizations (government research laboratories excepted). All others may obtain an individual license by choosing "Purchase..." from the Apple menu; a single-user license is \$25 US. Multi-user licenses are also available; for more information, visit our Web site.

### Contact Information:

Fetch Softworks  
PO Box 72  
Etna, NH 03750  
[info@fetchsoftworks.com](mailto:info@fetchsoftworks.com)  
<http://fetchsoftworks.com>  
Voicemail/FAX: +1 (603) 619-0397

## Where to start

If you are familiar with file transfer in general and FTP in particular, start with the "Quick Guide to Fetch" topic. If you are new to file transfer and would like to learn in a step-by-step fashion, try "A Fetch Tutorial." All of the Fetch commands and screen elements are explained under the "Reference" heading, and most computer terms used elsewhere in the help section are defined in the "Glossary of Terms" section.

Balloon Help is also available for most of the menus and buttons in Fetch (except on Mac OS X); see the "Help" topic for additional information.

## Licensing

Multiple-user and Internet Service Provider (ISP) licenses are available from Fetch Softworks. For more information, visit <http://fetchsoftworks.com>.

## Quick Guide to Fetch

### Connecting

When you start Fetch or choose the **New Connection...** menu item, you will be prompted with the **New Connection** dialog box. You must specify the name or IP number of the machine you wish to connect to (the "host"), either by typing it, or by choosing one from the Shortcuts pop-up menu. If required, enter a user ID and password. In order to use "anonymous FTP", the host name will suffice; Fetch will substitute the user ID "anonymous" and send a default password. NOTE: The information entered in the **New Connection** dialog box has nothing to do with your license to run Fetch. It specifies the server with which you are going to exchange files. If you need help collecting this information, contact your service provider (i.e. the company that runs the server in question).

### Directories

Fetch displays directories much like the Mac OS Finder displays folders in list view. Once you are connected, the current directory is displayed as the title of a pop-up menu above the file list. You can change directory in several ways. Double-clicking the name of a directory in the file list (or selecting it and pressing Return or Enter) causes Fetch to navigate to that directory. The pop-up menu above the file lists lets you move to any ancestor of the current directory. Choosing the **Change Directory...** command from the **Directories** menu prompts you for a directory name and, assuming you enter a valid directory name, and you have permission to view that directory, navigates to that directory. Finally, the bottom of the **Directories** menu lists directories you have visited on this host. You may quickly return to any of these by selecting it from the menu.

### Getting a File

To get a file, drag its icon to the Finder, double-click on its name, or select it and click the **Get File...** button. If **Automatic** download mode is selected, Fetch will try to figure out what kind of file it is from its name (see **Automatic/Text/Binary** topic in the **Reference** section). If the **Text** download mode is selected, the file will be treated as text, and you will have the option of converting any BinHex files embedded in the text file. If the **Binary** download mode is selected, Fetch will check to see if the file is in MacBinary III or AppleSingle format, and otherwise save it without any formatting or conversion. If you know what kind of file you are getting, select the appropriate mode. Most of the time, however, the **Automatic** setting will do the right thing.

### Getting Multiple Files

To get multiple files, or directories, click on them while holding down the Command key and then drag them to the Finder or click the **Get...** button.

### Putting a File

To transfer a file from your Macintosh to another machine, click the **Put File...** button. You will be asked to pick a file and prompted to name the uploaded file. "**Automatic**" format mode is the recommended default setting, as it will usually pick the correct format for the particular file.

### Putting Multiple Files

To transfer multiple files, or folders, to another machine, hold down the Option key while clicking the **Put File...** button, or choose **Put Folders and Files...** from the **Remote** menu.

### Cancelling and Quitting

You may cancel any ongoing operation by clicking the **Stop** button. Some host computers do not recognize the cancel command and will therefore continue the current operation until it is completed. In that case, your only options are to close the connection (by closing the transfer window) or quit Fetch by choosing the **Quit** command from the **File** menu.

## Why Transfer Files?

There are two main reasons to transfer files between a Macintosh and another computer. One is so that they can be used by programs on the other machine. The most common example of this is the World Wide Web, a global collection of files that can be accessed from anywhere on the Internet, using just about any kind of computer. The most common type of file shared this way is the text file (see the "Glossary" topic for a definition of text file; HTML files are text files). Other cross-platform file types include image files in formats such as GIF, PNG, Photoshop, EPS, Illustrator and JPEG; movie files in QuickTime and MPEG format; audio files in WAVE, MP3 or AIFF format; files created by productivity applications, such as Microsoft Office; Java applets; and compressed archives in GZip format. These files can be transferred to a Macintosh using **Binary** mode and to other computers in the **Raw Data** format (see "Raw Data" under "Reference"). Typically, you would use Fetch to transfer a file to a server computer, from which it could be accessed by others.

The other reason is to use the other computer as a holding place for Macintosh files. This sort of use can be termed "archiving." In this case, it is not important for the file to be in a format understood by the other computer, since it will only be transferred back to a Macintosh. It is important to be able to retrieve the file exactly as it was saved. Fetch supports three file formats that are suitable for archiving Macintosh files: MacBinary, AppleSingle and BinHex. A file stored in any of these formats can be reconstructed without any loss of information (for instance the file type, icon, and original name will be preserved).

This sort of file transfer can be used to distribute software over a large area. There are several large archives of Macintosh programs on the Internet, which can be downloaded by anyone with Internet access. The files are typically stored in MacBinary or BinHex format.

## A Fetch Tutorial

The following help items walk you through the process of connecting to a remote computer, navigating around its directories, and moving 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 under the "Glossary" help topic. If you are only interested in transferring web pages to a server, you can skip to the "Uploading Web Pages" tutorial topic.

## Connecting and Signing on

When you first open Fetch you are presented with the sign-on dialog box; at other times you can choose **New Connection...** from the **File** menu in order to see that dialog. The sign-on dialog has fields for entering three pieces of information:

- the name of the machine you wish to transfer files with (or "host name")
- the name of your account on that machine (or "user ID")
- the password for that account

You will usually be entering information about an account you have with a service provider, such as your Internet Service Provider or a web hosting service. You would get the host name, user ID and password from that service provider. Fetch Softworks is not a service provider.

The sign-on dialog also contains a shortcut mechanism for filling in some or all of these fields. To the right of the word "Shortcuts", there is a pop-up menu.

Example: Click the icon to use the menu and select the item titled "Fetch Softworks." When you release the mouse, the machine name field will be filled in. The machine name is "ftp.fetchsoftworks.com," a Unix server. Note that the user ID and password fields are blank. Some machines, including ftp.fetchsoftworks.com, allow any user to transfer files with it as a guest, without setting up an account or specifying a password. You may then click the **OK** button to initiate the connection.

## Finding Files and Directories

Most computers divide their files among directories, which are equivalent to folders on the Macintosh. Unless the file you want to download is at the first level directory - the "root" directory (denoted by a "/" - you'll have to navigate to the directory where the file is located. The current directory is displayed in the pop-up menu below the window's title bar. The list contains the names of the files and subdirectories for that current directory.

Example:

If you open a connection to `ftp.fetchsoftworks.com`, your current directory will be "/", and you'll 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 directory (another word for folder) in several ways. The simplest is to double-click on the name of a subdirectory in the list. You can try this by double-clicking on the name "example" in the current file list. The current directory will be "example" and the file list will show the names in that subdirectory. If you click on the name of the current directory, the pop-up menu displays the complete hierarchy leading to the current folder (quite like clicking on a Finder window while pressing the Command key). Select an item in the menu to change to make that the current directory.

## Getting a File

The current directory should be "example" (its complete pathname is "/example"). One of the files listed should be a GIF image named "fetchicon.gif."

In order to move "fetchicon.gif" from ftp.fetchsoftworks.com to your Macintosh, you can double-click its name. Alternately, you may select its name and press Return or Enter, or you can click the **Get File...** button. You will be prompted to name the file; if you click OK, the file will then be saved to your disk, complete with its icon, and ready to be used.

If you are using System 7.5 or later, you can also drag the selected file to a folder or hard disk on the desktop (or to the desktop itself). Using drag-and-drop lets you avoid naming the file and specifying its location in the file save dialog.



## Putting a File

In order to put a file on an FTP server, you must have "write" permission (i.e. the permission to create, rename, and delete files). Most anonymous FTP sites do not give guests write permission, so you will probably want to use a personal account for this part of the tutorial. [Of course, if you do not have any place to put files, you do not really need this part of the tutorial. (-)]

To upload a file to your FTP server, first enter the hostname, account name and password in the **New Connection** dialog and connect to the server. Then click on the **Put File...** button. You will be prompted to select the file you wish to send; since this is just a tutorial, you can pick any file, though you probably want to avoid very large files. You will then be asked to provide a name to give the file on the server. This dialog box also gives you a chance to pick a file format from a pop-up menu. Fetch will usually pick an appropriate name and format so you can simply press the **OK** button.

Alternately, if you are running System 7.5 or later and you can see the Finder icon for the file you wish to upload, you can simply drag the file's icon to the Fetch transfer window. In that case, Fetch will use the upload file format indicated in the transfer window.

The file will be transferred to the server and you will be able to watch its progress in the status panel on the right side of the Fetch 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 choose **Delete directory or file...** from the **Remote** menu (or press Command-Delete).

Congratulations -- you have just transferred files to and from your Macintosh! You can learn more about Fetch by consulting the "Reference" help topics. Balloon Help can also help you learn about the various Fetch objects and menu items (not available in Mac OS X). To enable Balloon Help, select **Show Balloons** from the **Help** menu, then point to objects and read the associated Balloon Help text.

If you have any comments, questions, or criticisms about this documentation or the Fetch application, please send mail to [feedback@fetchsoftworks.com](mailto:feedback@fetchsoftworks.com).

## Uploading Web Pages

One common use for Fetch is transferring files to web servers. Web authors often create their HTML, GIF, JPEG, etc. files on a Macintosh, and use Fetch to transfer the pages 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. This tutorial section is aimed at walking you through your first such transfer. You can read the other tutorial sections, but they are not required.

Before you start, you will 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.acme.com". Note that you don't want an entire URL (such as "http://www.acme.com"), just the hostname part.
- 2) Your user ID, the name under which you sign onto the server. Your user ID is usually a sequence of letters and/or numbers (such as "jmatthews"), and may be the characters to the left of the "@" in your e-mail address. Not to be confused with an email address, your user ID will almost certainly not include an @ sign.
- 3) Your password. Again, this may be the same password you use for e-mail and/or dialing your service provider.

If you do not have these pieces of information, the company that sold you the web server space (or the university that provided it for free) should be able to provide them. Or, you can check a list of links to various service provider help pages:

<http://fetchsoftworks.com/fetch.ispinfo.html>

Additionally, to check your uploaded web site, you'll need to get the URL from your service provider.

Before you start, if you are uploading a web site 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 relationships between files correct. Moving files to a different folder will probably break some links and image tags; make sure they all work before uploading.

To start, choose the **New Connection...** command from the **File** menu. Enter the three pieces of information in the appropriate text fields (you can leave all other fields blank), and click the **OK** button. If everything goes well, you will see a list of the files in your web directory. Otherwise, you will need to correct one or more of the pieces of information.

If you've successfully logged in to the server and are viewing the directory with your Web files, you can choose **New Shortcut...** from the **Customize** menu to create a shortcut for future use. This shortcut will be appended to the shortcuts list; to use, select the Shortcuts pop-up menu in the **New Connection** dialog (or the **Open Shortcut** menu command from the **File** menu).

Once you are connected, choose the **Put Folders and Files...** command from the **Remote** menu to upload your web page files. If all of the files are in one folder, you can navigate to that folder and use the **Add All** button to add them to the list of items to be uploaded. Otherwise, you can choose them one by one. You can also add folders to the list of items to be uploaded. Next you will be asked what format to use for the transfer; choose "**Automatic**", then click the **OK** button to begin the transfer.

Once the pages have been uploaded, you should check them with a web browser, such as Netscape Navigator™ or Internet Explorer®. You will need to know the URL for your main page; your web service provider should be able to provide this information. Enter the URL in the browser's 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 right file references. Also, be aware of the particular requirements of your web service provider. Some require that your main page be named "index.html", and some require uploading everything to a directory named "web" or "public\_html." CGI scripts may need to go in a particular directory, and you may need to set their execute permission bit (see the **Set Permissions** help topic).

## Reference

The following topics offer more in-depth information about Fetch and its capabilities. There is roughly one topic for each button, menu item, and supported file format. If you can not find some topic here, please send electronic mail to [feedback@fetchsoftworks.com](mailto:feedback@fetchsoftworks.com). We appreciate your input.

## AppleSingle

AppleSingle is Apple's official format for representing a Macintosh file on other types of machines. AppleSingle files have no distinguishing filename suffix, so it is usually necessary to select the **Binary** download mode in order to download AppleSingle files.

Fetch can get files in AppleSingle version 1 and 2 formats, and put files in AppleSingle version 2 format. Automatic conversion of AppleSingle files can be turned on and off in the Preferences dialog box.

## Automatic Upload Format

The "Automatic" upload format, like the "Automatic" download mode, tells Fetch to use the information available to it to make a choice between a number of alternatives. In the case of upload formats, the alternatives are the formats that Fetch supports: Text, Raw Data, MacBinary III, BinHex and AppleSingle. When the "Automatic" format is chosen, Fetch will try to do the "right thing" by following these rules (in order):

- If Fetch is connected to a server that can decode MacBinary files, use MacBinary III format. That way, the file will arrive without any missing information.
- If Fetch is connected to a server that can decode AppleSingle files, use AppleSingle format. That way, the file will arrive without any missing information.
- If the file name ends in .htm or .html, send it in Text format, so that line endings will be converted.
- If the file name ends in .gif, .jpg, or .jpeg, send it in Raw Data format, so that no conversion will happen.
- If the file name has a suffix, look up the suffix in Suffix Mapping and choose between Text and Raw Data formats, depending on the setting in Suffix Mapping. If the suffix is not found in Suffix Mapping, use Raw Data format.
- If the file type is found in Internet Config, use the format specified there: Text, Raw Data, or MacBinary III (if the Macintosh box is checked).
- If the file type is "TEXT" (the file type of text files), send it in Text format, so that line endings will be converted.
- If the file type is "APPL" or "appe" (the file types of applications) use MacBinary III format (because the file is only useful on Macintoshes).
- If the file has a resource fork, send it as MacBinary III so that the resource fork will be preserved.
- Otherwise, use Raw Data format.

## Automatic/Text/Binary

FTP (the protocol which Fetch uses to transfer files) divides files into two categories: text files and binary files. Furthermore, the Macintosh uses complex files that can only be represented on other computers with special formats. In order for a file transfer to work, these issues must be worked out. Fetch has three download modes that control the way a file is retrieved: they are titled **Automatic**, **Text**, and **Binary**. It would be convenient if Fetch could always tell what sort of file it was getting and treat it in the appropriate way; unfortunately, Fetch is not that clairvoyant. So some of the time you will 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 setting will be the right one most of the time. When this mode is selected, Fetch tries to interpret the name of the file you are getting. For instance, most BinHex filenames end in "hqx," and most GIF (Graphics Interchange Format) files end in "gif." When the **Automatic** mode is selected these clues lead Fetch to download "hqx" files as text files (and ignore the non-BinHex part) and download "gif" files as binary files (with the right icon to be opened by a popular GIF viewing program). These clues are managed by the **Suffix Mapping...** menu command (see the "Suffix Mapping" help topic).

Some filenames will not offer any clues, and in that case Fetch will use its default behavior. The default behavior can be changed in **Preferences**.

When Fetch believes a file is in BinHex format, it discards all non-BinHex data and only allows you to save the converted file. This is a problem if you want the other data or if the file is only part, and not the first part, of a complete BinHex file.

When Fetch thinks a file is binary information, it first checks to see if it is in MacBinary or AppleSingle format. If it is, it will be decoded and put back in its original state (unless that option is turned off in Preferences); otherwise the type and creator (which specify the icon to use) will be looked up in the Suffix Mapping list.

The Automatic setting may do the wrong thing if the filename does not reveal the file type, or suggests the wrong file type (e.g. a ".txt" file that is in fact binary). For these situations you must use one of the other two settings.

**Note:** There are two notable exceptions to these rules for Automatic mode. If Fetch determines that the server machine is a Macintosh running Mac OS, it will try to transfer all files in MacBinary mode. This way, all files will appear exactly as they do on the server. Similarly, if Fetch determines that the server machine is a Macintosh running MachTen (a Unix variant), it will try to transfer all files in AppleSingle format, since AppleSingle is MachTen's preferred format for Macintosh files.

### Text

This setting tells Fetch that the file in question is a text file (or a BinHex file, which is one kind of text file). Since the Automatic setting usually treats mysterious files as text files, this setting is rarely necessary. If, however, a text file has the sort of name usually associated with binary files (i.e. ending in "bin", "Z", "sit", "gif", or "tar"), you should choose this setting to override the automatic behavior. Likewise, if a filename ends in "hqx" but does **not** refer to a complete BinHex file, or the first part of a complete BinHex file, you should choose this setting.

### Binary

This setting is for getting binary (i.e. 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 (e.g. it is in MacBinary or AppleSingle format), choosing this setting will help ensure the proper treatment. When this mode is set, the file will be checked to see if it is in MacBinary or AppleSingle format.

## BinHex

BinHex (sometimes called BinHex4) is a format for representing a Macintosh file in text form. The Macintosh file is converted to a series of lines, each made up of letters, numbers, and punctuation. 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 represented some other way. The suffix ".hqx" usually indicates a BinHex format file.

Fetch can get and put files in BinHex format. Conversion of BinHex format files can be turned on and off in the **Preferences** dialog box.



## Change Directory

The **Change Directory** command is found in the **Directories** menu. It allows you to specify the current directory by typing its full path, rather than navigating through the hierarchy one directory at a time. You can also type or paste a Uniform Resource Locator (URL) into the **Change Directory** dialog box. This lets you move to a directory on any server, even if you're not currently connected to that server. See the "Uniform Resource Locators" topic for more information on URLs.

## Contextual Menus

Fetch supports the contextual menu feature (Mac OS 8.0 and later), which provides convenient access to often-used commands with 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, view, edit, rename, delete, set permission, etc.

To display the contextual menu, press the Control key and click an item; contextual menus are available in Fetch transfer and shortcut windows.

There are two commands that are currently 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 a file on a server. Duplicating remote files is done as a sort of server-to-server transfer (see the **Server-to-Server Transfers** topic).

## Create host directory

This menu command creates a new directory on the remote computer. You will be prompted for a name to give the directory; unless you specify otherwise the directory will be a sub-directory of the current one. You should only use this command if you have the appropriate privileges and the remote computer supports directories. If either of these conditions are not met you will see an error message.

## Delete directory or file

This menu command allows you to delete files and directories on the host computer. If you select a name from the file list and choose this command, you will be asked for confirmation before the file is deleted. If you have selected more than one file and give confirmation, all the files will be deleted. If you have not selected a file, you will be prompted for the name of the file to delete and then it will be deleted. Note that you must have the necessary privileges to delete the file(s) in question. Otherwise, you will see an error message and the file(s) will remain unchanged.

If you select a directory and choose this command, Fetch will delete the directory's contents before deleting the directory itself.

In System 7.5 or later, you can also delete remote items by dragging them to the Finder Trash icon.

## Directories Menu

There are two menus to make moving between directories more convenient:

- There is a pop-up menu above the file list on the left side of the Fetch window. The title of this menu is the name of the current directory, or "<unknown directory>" if Fetch was unable to determine the name of the current directory. This menu includes the current directory and the entire hierarchy leading to it. Choosing one of these directory names makes it the current directory and updates the file list.
- The **Directories** menu in the menu bar includes the names of directories on the remote computer that Fetch has visited. Choosing one of these directory names makes it the current directory and changes the file list. To clear this list, choose the **Clear Directories Menu** command.

## Drag and Drop

Beginning with System 7.5, the Mac OS includes drag-and-drop capabilities for moving information within and between Macintosh applications. Fetch takes advantage of this facility 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 a remote file or directory, drag it from a file list or shortcut list to a Finder icon, window, or the desktop.
- To delete a remote file or directory, drag it from a file list to the Finder or OS X Dock trash icon.
- To upload a file or folder, drag it to a transfer window or directory shortcut (in a shortcut list).
- To move a file or directory into a subdirectory, drag it to the subdirectory icon.
- To move a file or directory up a directory level, drag it to the open-folder icon above the file list.
- To copy a file or directory into a subdirectory, drag it to the subdirectory icon with the Option key pressed.
- To copy a file or directory up a directory level, drag it to the open-folder icon above the file list with the Option key pressed.
- To transfer a file or directory 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 create a shortcut for the current directory drag the open-folder icon to a shortcut list window.
- To create a shortcut file for the current directory drag the open-folder icon to a Finder icon, window, or the desktop.
- Dragging shortcuts between shortcut list windows moves them (or copies them if the Option key is held down).
- To delete a shortcut from a shortcut list drag it to the Finder or OS X Dock 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).
- Selected text can be dragged from text windows.
- To specify the download folder, drag a Finder file or folder icon to the Preferences checkbox.
- To specify the default type and creator bytes, drag a Finder file icon to the Preferences text fields.
- To specify your preferred text editor, drag its icon or the icon of one of its documents to the Preferences text editor pop-up menu.

## Edit File

Use the **Edit File with BBEdit** and **Edit File with GraphicConverter** commands to edit files on an FTP server using BBEdit (version 4.0 or later, from Bare Bones Software: <<http://www.barebones.com>>) or GraphicConverter (version 3.8 or later, from Lemkesoft, <<http://www.lemkesoft.com>>). Text files (including HTML) should be edited with BBEdit, image files with GraphicConverter. Fetch will download the file and use the appropriate application to open it. When you use the "Save" command to save your changes, the application will instruct Fetch to upload the changed file to its original location. This provides a convenient, automatic way to edit files on a server.

If no file is selected, you can use the **Edit New File with BBEdit** and **Edit New File with GraphicConverter** commands to create new text or image files, respectively, which will be uploaded to the server when you save them.

Note that you must leave Fetch running while you edit files in BBEdit or GraphicConverter (you do not have to remain connected to the server, however). Also, this feature requires recent versions of BBEdit or BBEdit Lite (4.0 and later) and GraphicConverter (3.8 and later).

Some other editors (such as Style and Tex-Edit) also support the external editor protocol used for this feature. You can change what editor Fetch uses with the **Obscure Options** button in the **Misc** section of **Preferences**.

## File List

The file list is the list of file and directory names on the left side of a Fetch transfer window. It works much like the file list in the standard Macintosh "Open" dialog box. Clicking on a name selects it, and many operations (Get file, Delete file or directory, Rename host file, etc.) operate on the selected name. Holding down the Command key allows you to select multiple individual filenames, and holding down the Shift key while clicking selects a range of filenames. 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 on a name prompts Fetch to change directory to the selected name. If the name refers to a file, the "change directory" command should fail, and Fetch will then download the file. Pressing Return or Enter is equivalent to double-clicking.

Fetch displays a small icon next to each name. Files and directories are designated with the standard file and folder icons. If Fetch can not determine whether the name refers to a file or folder an icon showing a question mark is displayed. If the name refers to a Unix symbolic link (an alias for a file or directory in a different place) it is displayed in italics (just as the Finder displays the names of aliases on the Macintosh).

Fetch also displays a small icon of an open folder above the file list. If you are using System 7.5 or later, you can drag that icon to the Finder or a shortcut list to create a shortcut for the current directory, or drag item icons to it in order to move them up a directory level.

The file list can be sorted by name, kind, size, or date. The current sort order is indicated by the underlined or darkened column title. To change the sort order, click on a different column title. On Mac OS 8.1 and newer, the columns can be resized by clicking between the titles and dragging.

Fetch stores file lists in the "Fetch Cache" file in the Preferences folder within the System Folder. Fetch will re-use file lists rather than downloading them when there is reason to believe that the stored list is up to date. [The technical details: a file list is considered "good" for 5% of the age of its newest item, so if latest date in the list is 20 days old, the list will be considered valid for the following day]. When Fetch uses a stored file list, it prints the date and time when the list was originally retrieved below the file list. Sometimes the stored list will be out of date, in which case you can reload it from the server by choosing **Refresh File List** from the **Directories** menu.

Note for experienced users: Pressing down arrow while holding down the Command key makes the selected directory name the current directory; pressing Cmd-up arrow moves to the parent directory. Users with appropriate keyboards can use the Home, End, Page Up, and Page Down keys to scroll the file list.



## Find

Fetch can search for information in any of its windows (except for media windows) with the **Find** command. When a Fetch transfer window is frontmost, **Find** searches through the displayed file names for the desired text. When a text window (including the **Fetch Transcript** window) is in front, **Find** searches the displayed text. When a shortcut list window (including the **Fetch Shortcuts** window) is in front, **Find** searches the names and URLs of the shortcuts. When the **Fetch Help** window is frontmost, **Find** searches through all the help topics.

## 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. FTP was developed in the 1970s by users of the Advanced Research Projects Agency network, or Arpanet. FTP is the official file transfer mechanism of the Internet (Arpanet's larger successor) and therefore is supported by millions of machines.

Fetch implements one part of FTP, the "client" side. Therefore, it can communicate with any connected machine that has a "server" program. Some of Fetch's features will not be available on certain servers -- this is because the other machine does not support the feature (e.g. not all machines have sub-directories) or because the server program does not implement the complete FTP specification.

## Get Directories and Files

The **Get Directories and Files** command is used to download multiple files and/or directories with one command (you can also download files and directories by pressing the **Get...** button). You can select the files or directories you wish to download in the file list; if none are selected, you will be prompted for a file or directory name. Next, you will be prompted for the folder in which to save the files and directories on your Macintosh disk.

You can also type or paste a Uniform Resource Locator (URL) into the **Get Directories and Files** dialog. This lets you download a file or directory on any server, regardless of whether you are currently logged in. See the "Uniform Resource Locators" topic for more information on URLs.

The names of the downloaded files will be set automatically, based on the names that the files have on the remote computer. All format conversions will also be performed automatically, according to the settings in the **Preferences** dialog.

The progress of the file transfer, including the name of the file being transferred, is shown in the **Status** panel. You can cancel the transfer by pressing the **Stop** button.

## Get File

The **Get File...** button is used to retrieve files from the remote computer. Fetch will attempt to download the selected file. If no filename is selected or the Option key is held down, Fetch will instead prompt the user for a filename. This can be useful if the desired file is in a different directory.

You can also type or paste a Uniform Resource Locator (URL) into the dialog. This lets you download a file from any server, even if you're not currently logged in. See the "Uniform Resource Locators" topic for more information on URLs.

Once the download has been started, the file will either be stored under a default name in the download folder specified in **Preferences**, or you will be prompted to name the file to save. The progress of the download will be charted in the transfer window **Status** panel. Pressing the **Cancel** button will stop the transfer.

If more than one filename is selected, the button will be titled **Get Files...**, and you will be prompted for the folder into which the files will be downloaded. Similarly, if a directory is selected, the button will be titled **Get...**. The progress of the download (including the name of the file currently being transferred) will be indicated in the **Status** panel. Multiple filenames can be selected by holding down the Command key while clicking on the names.

The success or failure of a file transfer will often depend on the settings of the download mode pop-up menu. See the "Automatic/Text/Binary" help topic for more information on those choices.

## Get Info

The **Get Info** command in the **Remote** menu displays information about the selected file or directory. In the case of directories, Fetch counts the files and directories within the directory and computes the amount of disk space being used in total. This can be used, for example, to determine how close you are to your server storage limit. For files, the **Get Info** command displays the file's icon, kind, size, URL, modification date, download mode, and permissions, and lets you edit the file's name.

## Help

The Fetch Help menu command opens the Fetch Help window (this window) and makes it the active (frontmost) window. Click a help topic in the list on the left and scroll through the help information on the right using the scroll bar. You can copy information from this window, as well as print the current topic.

Additional help for most user interface elements - menu items, window and dialog box controls - is available in System 7 through Mac OS 9 in the form of Balloon Help. In the Help menu, select the Show Balloons menu item. Move the mouse pointer to identify an element and see information about that item.

Close the Help window by clicking its close box or press Cmd-W.

## Internet Config

Mac OS System 8.5 through 9.x include the Internet control panel, which allows various Internet-related settings to be configured from one interface. These settings are then available to all Macintosh programs, including Fetch, email applications and Web browsers. The Internet control panel can be used in basic mode to configure identity (name, email, company), email (login, password, servers) and Web browser preferences. In advanced user mode, the **Advanced** tab appears; it allows more detailed settings, including "helper" applications, where you can specify the preferred text editor and ftp application (we hope you'll select Fetch), and file suffix mappings.

The Mac OS Internet control panel evolved from the Internet Config system, created by Quinn "The Eskimo!" and Peter N Lewis. The venerable Internet Config can be installed on earlier versions of the Mac OS that don't have the Internet control panel.

If the **Use Internet Config** checkbox is selected in **Preferences**, Fetch uses the Internet (or Internet Config) control panel for the following settings:

- Text editor
- Default binary type and creator
- Download folder
- Default text font and size
- Default shortcut list font and size
- External text and image editors
- Passive mode transfer setting (except on Mac OS X)
- Proxy server and SOCKS gateway settings (except on Mac OS X)
- Suffix mappings (except on Mac OS X)
- Post-processing actions (except on Mac OS X)

You can edit all but the last two of these from within Fetch; Internet Config's suffix mappings and post-processing actions can only be edited with the Internet Control Panel or the Internet Config application (which Fetch will launch for you if you choose **Suffix Mapping...** or **Post-Processing...** from the **Customize** menu).

In Mac OS X, some of these settings can be edited in the **Network** pane of **System Preferences**. At the time of this Fetch software release, there is no user interface for editing suffix mappings or post-processing actions. (See the "Suffix Mapping" and "Post-Processing" help topic for additional information.) For that reason, starting with version 4.0.2 Fetch uses its own suffix mappings and post-processing actions on Mac OS X even when **Use Internet Config** is checked.

One other feature of the Internet (Config) control panel is the ability to open a Uniform Resource Locator (URL) from within a document, such as this help page. Position the mouse over some part of the URL, hold down the Command key and click on the URL. This will open the URL with the specified helper application.

## ISO Translation

Many computers, including the Macintosh, can display special characters that are important for communicating in Roman languages besides English (some examples: é ö å ç). Unfortunately, different computers use different codes to refer to these characters. So, unless special translation is performed, a filename or text file that looks fine on one computer can be a garbled mess on another.

Fetch provides the **Translate ISO Characters** option in **Preferences** to address this problem. When this option is enabled, Fetch will translate special text characters between the Macintosh standard and the ISO 8859-1 character code, a standardized code set for these characters. It has been adopted by some computer manufacturers, and others support it as a "neutral ground" between incompatible codes. If Fetch is connected to one of these machines, the special characters should translate perfectly.

Note that Fetch only translates text file contents and filenames. Binary files (such as compressed text files) are not translated, and text files must be transferred in Text (or Automatic) mode for translation to occur.

The ISO-to-Mac and Mac-to-ISO translation tables are stored in 'taBL' resources with IDs 1001 and 1002, respectively. If you need to translate with a computer that uses some character set besides ISO 8859-1, you can edit these tables. The two tables should be the inverse of each other, so that an upload-download cycle results in the same file that you started with.

NOTE: The **Translate ISO Characters** option will garble texts in character sets other than ISO 8859-1 (such as ISO 2022-jp), and so should be turned off when transferring texts in those character sets.



## Keychain

In Mac OS 9, Apple introduced a password storage facility called the "Keychain". It lets users store their various passwords in a central place, for more convenient security. Fetch supports this feature when it is available. When you type a hostname and user ID, and your keychain is unlocked, Fetch will check it to see if you've saved a password for the server and account in question. If you have, that password will be automatically entered in the **Password** field of the **New Connection** window. Similarly, you can open a shortcut that does not include a password as long as the keychain is unlocked and contains the password. When you enter a password, you can select the **Add to Keychain** checkbox to have that password added to your Keychain for future use.

## MacBinary III

MacBinary III (sometimes simply called 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 III is a compact format but cannot be passed through most electronic mail systems. MacBinary files typically have names ending in ".bin".

Fetch can get and put files in MacBinary III format, as well as MacBinary I and MacBinary II format (earlier versions of the MacBinary standard). Automatic conversion of MacBinary files can be turned on and off in the **Preferences** dialog box.

Some FTP servers (notably Stairways Software's NetPresenz and Apple's AppleShareIP) recognize MacBinary format. If Fetch determines that it is connected to one of these servers, it will try to use MacBinary III for all transfers.

## Mirror Folder

The **Mirror Folder...** command in the **Remote** menu lets you copy all the new or changed files from your Mac to a server, or vice-versa. For example, a web site author may keep all the files that make up a web site in a certain folder on her Macintosh. The **Put Folders and Files...** command can upload the entire site, but if only a few files have changed since the last upload it isn't necessary or efficient to upload the entire site. Instead, the **Mirror Folder...** command can be used to upload only files that are new or have been changed. You can also save a "mirror document" that describes a Mirror Folder command, and simply open that document in order to perform the mirror operation. That makes uploading web site changes a simple one-click process.

When you give the **Mirror Folder...** command, Fetch presents a dialog box where you specify the source and destination for the mirror operation. They can both be remote directories (in which case Fetch will perform server-to-server transfers), or one local folder and one remote directory. You can specify a local folder with the standard "choose folder" dialog, or by dragging a Finder icon to the dialog box. Remote directories can be chosen from a list of currently open directories, or entered in a window similar to the New Connection dialog box. Arrow buttons indicate which way the files will be copied. "**Delete stray items at destination (Caution!)**" instructs Fetch to delete items in the destination folder that do not exist at the source; this is a dangerous command, since it can delete entire directories.

The **Save...** button lets you save the specific mirroring command to a file. Opening that file will recreate the mirror dialog box, or (if you choose) immediately perform the operation.

## New Connection

The **New Connection** menu command brings up the **New Connection** dialog box. Enter the name or IP number of the server you would like to connect to, your account name (user ID), and the associated password. If the account name is omitted, the user name "anonymous" will be assumed; this user name is used for guest access to an FTP site, also known as "anonymous FTP". When connecting as "anonymous" it is not necessary to enter a password.

NOTE: The information entered in the **New Connection** window has nothing to do with your license to run Fetch. It specifies the server with which you'd like to transfer files. Your Internet Service Provider (ISP) should provide you with information to connect to their FTP server.

If you have installed the necessary software, you can choose security options from the **Security** pop-up menu. For more information, see the "Security" help topic.

You can fill in some or all of these fields by choosing an item from the Shortcuts pop-up menu. Click the icon at the bottom of the signon dialog and the menu will pop-up. (In Mac OS 8.1 and earlier, you can also cycle through the shortcuts by pressing Cmd-Tab, or go backwards by pressing Cmd-Shift-Tab.) For convenience, you can specify one of the shortcuts as the default shortcut; in that case, the **New Connection** window will always open with information from the default shortcut already filled in. To create a new default shortcut, fill in the information and choose **Create Default Shortcut** from the Shortcuts pop-up menu.

In Mac OS 9 and later, you can save your passwords in the Keychain by clicking the **Add to Keychain** checkbox before pressing the **OK** button. For more information, see the "Keychain" topic.

If you click the triangle, the window expands to reveal additional fields. For convenience, you can enter a specific directory on the server, so that Fetch automatically navigates to that directory when you open a connection to that server. If the server uses a non-standard TCP port, you can enter this number in the appropriate field. If you'd like Fetch to repeatedly try to connect to a busy server, you can specify the number of retries in that field.

The signon dialog box supports "smart paste" of Uniform Resource Locators (URLs). If you paste a URL, it will be broken into parts and the parts placed in the appropriate fields. See the "Uniform Resource Locators" topic for more information on URLs.

You can open up to sixteen connections at a time. Each connection uses memory, so if you wish to open more than three or four, you should increase Fetch's memory allocation in the Finder.

## One-Time Password

One-Time Password (OTP) is a challenge-response password system that is expected to become an Internet standard. Servers running the 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 re-used, 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.

Fetch offers automatic support for the OTP system. If Fetch detects the 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. To be warned in this case, check the **Warn before sending password insecurely** box in the Security section of Preferences.

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.

OTP is based on the older S/Key™ system developed by Bellcore (S/Key is a trademark of Bellcore), which Fetch continues to support.

## Open Shortcut

The **Open Shortcut** menu lists the shortcuts from the **Fetch Shortcuts** window and allows you to connect to one of the listed machines with one command. To edit the list of shortcuts, choose **Fetch Shortcuts** from the **Window** menu and use the **New Shortcut**, **Edit Shortcut**, and **Remove Shortcut** commands from the **Customize** menu. You can also paste URLs into the **Fetch Shortcuts** window (or any shortcut list window) and drag items from file lists into any shortcut list.

## Post-Processing

Fetch can automatically “pass-off” downloaded files to other applications; this feature is called post-processing.

If you have installed Internet Config, and have checked the **Use Internet Config** checkbox in **Preferences** (see the "Preferences" topic), Fetch uses the post-processing settings maintained by Internet Config (see the "Internet Config" topic). In that case you should open the Internet control panel (or, on older systems, the Internet Config application) to change the settings.

Otherwise, you can edit your post-processing settings with Fetch’s Post-Processing dialog. In that dialog you can create a list of post-processing actions. An action consists of two pieces of information: the kind of file that the action applies to, and the application that this kind of file should be opened with. The kind of file is specified by type/creator codes, and the application by its signature code. For instance, one action could specify files with the type code of “TEXT” and the creator code of “ttxx”, and the application signature of “MSWD.” This would mean that SimpleText files (which have type “TEXT” and creator “ttxx”) should be automatically opened with Microsoft Word (which has the signature “MSWD”). Alternatively, an action could specify a type of “TEXT” and a creator of “\*\*\*\*\*”; the special code “\*\*\*\*\*” matches anything, so this action would apply to all files of type “TEXT” (i.e. all text files). Finally, the application signature can be left blank. In this case the file will be opened by the application that typically opens it, just as if you double-clicked on the file in the Finder.

The codes that Fetch uses to identify files and applications are rather cryptic, so Fetch provides an alternative means for entering them. The **“Choose Example...”** and **“Choose Application...”** buttons allow you to select the kind of file or application you are interested in, leaving Fetch to extract the necessary codes.

Post-processing actions can be enabled and disabled. This allows you to define actions and turn them off temporarily, without having to re-enter them later. Fetch comes with several post-processing actions defined; they give some examples of how this feature can be used.

## Preferences

The Preferences menu item brings up a dialog that allows you to configure the way Fetch operates. The settings made in this dialog are saved in a file called "Fetch Prefs" in the Preferences folder located in the System Folder.

The following can be configured in the Preferences dialog:

### General Preferences:

**Use Internet Config** -- When this box is checked, Fetch will use the Internet Config system for storing preferences that are commonly shared between different Internet applications (these are indicated below by the statement "Can be set with Internet Config"). For more information, see the Internet Config help topic.

**Display server messages** -- Some FTP servers display special information in the client-server dialog, such as hours of operation, use restrictions, number of open connections, and information about the files on the server. If this box is checked, Fetch will display this information in a separate messages window; otherwise, they can be viewed by examining the Fetch Transcript window.

**Keep connections active** -- If this box is checked, Fetch will regularly send commands to the remote computer 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 service.

**Default Shortcut** -- If you've selected a default shortcut, the "New Connection..." dialog box will be automatically filled in with information from that shortcut.

**Background Notification** -- Choose the sound (if any) that you would like Fetch to make when it completes an operation while in the background.

**Blink Icon in Menu Bar** -- Have Fetch blink an icon in the application menu whenever it finishes an operation in the background. [On Mac OS X this setting is named **Bounce icon in Dock**, and as the name suggests it controls whether the Fetch icon bounces in the Dock whenever it finishes an operation in the background].

### Download Preferences:

**Default Download Mode** -- pop-up menu to specify the default download mode for new transfer windows. The options are: Automatic, Text or Binary.

**"Automatic" defaults to "Text"** -- This checkbox controls how Fetch will download a file when the "Automatic" download mode is chosen and the file name does not indicate whether it is a text or binary file. If the box is checked, Fetch will assume it is a text file; otherwise Fetch will assume it is binary. For more information, see the "Automatic/Text/Binary" help topic.

**Overwrite files with conflicting names** -- if this box is checked, Fetch will overwrite an existing Mac file if it is in the same folder and has the same name as the file being downloaded. Otherwise, Fetch will add a number to the file name of the new file to differentiate it.

**Use download folder** -- If this box is checked, Fetch will save files in the specified download folder without any further prompting. To disable this feature, deselect the checkbox. The name of the



download folder is displayed in the adjacent button. If you press that button, you will be prompted to choose a new download folder (alternately, on System 7.5 and newer, you can drag the folder's icon to the button). The download folder can also be set with Internet Config.

**Edit text files with** -- This pop-up menu allows you to choose which file type and creator will be assigned to downloaded text files (and, consequently, which icon those files will display). Thus, you can have your favorite editor start up when you double-click on the icon of the retrieved file. If your favorite editor does not appear in the menu, you can drag its icon (or an alias) to the menu (assuming you are running System 7.5), or choose the Other... item in the menu and select it in the open file dialog. Note that if a text file has a suffix that matches a Suffix Mapping entry, the Suffix Mapping will take precedence over this setting. Can be set with Internet Config (i.e. the mapping entry for the ".txt" suffix).

**Text File line endings** -- This pop-up menu lets you choose what style of line ending Fetch should use for downloaded text files. Macs have traditionally used a single carriage return (CR) to indicate a line ending. UNIX systems, including Mac OS X, use a single linefeed (LF). DOS and Windows systems use a carriage return followed by a linefeed (CRLF).

**Default binary type/creator** -- The type/creator codes entered here will be used for raw binary files that do not have a type/creator assigned by the suffix mapping process. This can be useful if you are downloading many binary files that do not have a common suffix. If you are using System 7.5 or newer, you can drag a file to these fields; they will be filled in with the type/creator of the dragged file. Can be set with Internet Config (i.e. the mapping entry for the ".binary" suffix).

**Recognize and decode** -- These checkboxes determine whether Fetch will automatically recognize and convert downloaded files in the given formats (BinHex4, MacBinary, and AppleSingle). If you have trouble downloading a file in one of these formats, try deselecting the conversion option, download the file and convert the file manually with StuffIt Expander or other utility.

#### **Upload Preferences:**

**Default Upload Format** -- This pop-up menu lets you choose the format to use for uploading files. You will still be prompted for "manual" upload operations, but this format will be used when you use drag-and-drop to upload a file to a server. If you choose "Automatic", Fetch will try to determine the right format to use by consulting Internet Config and examining the file in question.

**Add .hqx suffix to BinHex uploads** -- If this box is checked, Fetch will add the ".hqx" suffix to the names of files being uploaded in BinHex format. Using this common suffix makes it more likely that the file will be retrieved intact. You will have the chance to override the suggested name if you are uploading a single file.

**Add .bin suffix to MacBinary III uploads** -- If this box is checked, Fetch will add the ".bin" suffix to the names of files being uploaded in MacBinary III format. Using this common suffix makes it more likely that the file will be retrieved intact. You will have the chance to override the suggested name if you are uploading a single file.

**Add .txt suffix to Text uploads** -- If this box is checked, Fetch will add the ".txt" suffix to the names of files being uploaded in Text or Wrapped Text format. Using this common suffix makes it more likely that the file will be retrieved intact. You will have the chance to override the suggested name if you are uploading a single file.

**Add Internet Config suffix to Raw Data uploads** -- If this box is checked, Fetch will add the suffix listed in the File Mapping section of the Internet control panel (or the Internet Config

application) to the name of the file being uploaded in Raw Data format. Many systems determine the type of a file from its name suffix; for example, a JPEG file is handled better when its name ends in .jpg. You will have the chance to override the suggested suffix if you are uploading a single file.

**Encode and decode file and folder names** -- This checkbox determines how Fetch converts the names of local files and folders into remote names, and vice versa. If the box is checked, 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 deselected, 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).

### **Firewall Preferences:**

Note: these preferences let Fetch work around network restrictions that some sites impose for security reasons. You should consult with your network administrator or help desk before changing these preferences.

**Use passive mode transfers (PASV)** -- If this box is checked, Fetch will initiate data connections, rather than asking servers to connect back to your Macintosh. Some company firewalls prevent incoming connections; this option makes it possible to use Fetch behind those firewalls.

**Proxy type** -- Some networks only allow FTP sessions from a special machine running something called a proxy server. In that case, Fetch needs to connect to the proxy server, which in turn connects to the actual server. If your network has such a proxy server, you should choose the matching type from this pop-up menu. Your firewall administrator should be able to tell you what to choose, and what to enter in the other fields. Note: Fetch does not support CERN HTTP proxy servers.

**Don't use proxy for** -- Proxy servers and SOCKS gateways may not accept connections bound for local servers (i.e. servers that are inside a 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 machines in that domain. So, if you entered "acme.com", Fetch would not use the proxy server or SOCKS gateway for connections to "ftp.acme.com", "internal.acme.com", or "acme.com", but it would continue to use the server or gateway for machines that are not in the "acme.com" domain. You may enter multiple domain names, separated by commas, spaces, or carriage returns, if there are multiple domains inside your firewall (for example, your firewall might protect "research.acme.com" and "sales.acme.com" while "ftp.acme.com" is outside the firewall; in that case, you would enter "research.acme.com, sales.acme.com"). Can be set with Internet Config.

### **Miscellaneous Preferences:**

**Use "TYPE I" to transfer binary files** -- Check this box if you want Fetch to use the "TYPE I" format for binary transfers; this is necessary for proper operation with some IBM mainframe servers, but can cause problems with machines that do not use 8-bit bytes (such as TOPS-20 systems).

**Omit file names beginning with a period** -- Check this box if you do not want Fetch to display names that begin with a period in its file list. This can reduce the clutter when connecting to Unix servers.

**Translate ISO characters** -- Check this box if you would like Fetch to translate 8-bit ISO characters in text files and file listings to the Mac equivalents, using the 'taBL' resources in Fetch.

**Show sign-on dialog at startup** -- Check this box if you would like Fetch to automatically open the sign-on dialog box when Fetch is started.

**Use case-sensitive sorting** -- Check this box if you would like Fetch to consider case when sorting file or shortcut names, so that all the names starting with upper case letters will be separate from the others.

**Use folders first sorting** -- Check this box if you want Fetch to always list folders first when you sort file lists by name.

**Allow duplicate connections** -- If you check this box, Fetch will let you issue commands (e.g. to download a file, or change directory) while a transfer window is already busy. In that case, Fetch will open a new, duplicate connection to perform the command. This can be convenient, but it is easy to accidentally create new connections when you did not realize that the original connection was busy.

**Anarchie emulation mode** -- If this box is checked, Fetch automatically creates a shortcut list and fills it with shortcuts for the items in the file list whenever the file list is changed. This makes Fetch operate a bit like Anarchie, the FTP client by Peter Lewis. It can be convenient to view directories in shortcut list windows, since you can easily have multiple windows open to different directories. But some commands (such as View File List and Send FTP Command) only work with the transfer window in front, not with shortcut list windows.

**Treat mystery items as files** -- On some servers, Fetch can not tell files from directories in the file list. On those servers, Fetch will start a "Get" or "Delete" operation by trying to change directory to the name in question, in order to determine whether it is getting/deleting a file or a directory. But a small number of servers (including MVS and Tandem models) do not return an error when Fetch tries to change directory to a file name, so Fetch erroneously concludes that it is getting/deleting a directory and the operation fails. When this checkbox is checked, Fetch will instead assume that such mystery items are files, and Fetch will not try changing directory.

**Ignore file list cache** -- If this box is checked Fetch will always download a fresh file list when it enters a directory. Otherwise Fetch will try to save time by using a cached file list. When Fetch uses a cached file list it displays the date when the list was downloaded below the file list (e.g. "As of 10/15/2002 12:15 PM").

**Obscure Options** -- These options cover unusual situations that most users do not encounter.

#### **Security Preferences:**

**Warn before sending password insecurely** -- this option ensures that you will always be warned before Fetch sends your password unencrypted (e.g. because a server does not support Kerberos).

**Warn before sending cleartext password to secure server** -- this option ensures that you will always be warned before Fetch sends your password unencrypted to a server that supports security.

**Always check protection level on incoming data** -- this option is meant to catch situations where the server starts sending unencrypted data; however, at the time of this software release, servers do this quite commonly, so it is usually more convenient to skip this check.

**Make Kerberos v5 tickets forwardable** -- this option allows certain ftp servers to further authenticate you to other services which may be required, such as a network file system (e.g. AFS).

**Send dummy password to Kerberos servers** -- this option works around a bug in some servers available at this writing.

## Put File

The **Put File** button is used to transfer a file from your Macintosh to the host computer. When you press the button, you will be prompted for the file you wish to send. Then you will see a dialog box that lets you set the name the file will be given on the other computer and the format it will be stored in. Be sure to pick a name that will be acceptable to the other machine (some systems require file names that are shorter than Macintosh file names and most forbid the use of certain characters). Note that if a file of the name you specify already exists on the other computer, unless you check the "In case of name conflict, pick another name" box, it will be overwritten without warning.

The formats you can use when putting a file are Automatic, Text, MacBinary III, Raw Data, BinHex, and AppleSingle. The Automatic format is usually the best choice: it makes an intelligent guess as to the best format to use, based on the type of file being transferred. You can specify the default format to use in Preferences (see the "Preferences" topic). See the help topic for each format for more information.

The progress of the file transfer is shown in the **Status** panel. You can cancel the transfer by pressing the **Stop** button.

You can upload multiple files and/or folders by pressing the **Put File** button with the Option key down, or by choosing **Put Folders and Files** from the **Remote** menu. For information, see the "Put Folders and Files" help topic.

The keyboard shortcut for the **Put File** button is Cmd-T.

If you are running System 7.5 or later, you can drag files and folders from the Finder into transfer windows or directory shortcuts; this can be a more convenient way to start an upload. In that case, the format displayed in the transfer window (or the default format specified in the **Upload** section of **Preferences**) will be used.

## Put Folders and Files

The **Put Folders and Files** command is used to upload multiple files and/or folders with one command. You will be prompted for the files and folders you wish to send. Then you will see a dialog box that lets you choose the format to be used. The names of the uploaded files will be set automatically, based on the names that the files have on your Macintosh, with suffixes added according to the settings in the Upload section of Preferences. Note that if a file of the same name already exists on the other computer it will be overwritten without warning, unless you check the "In case of name conflict, pick another name" box.

If you choose to upload a folder, Fetch will create a directory on the remote computer and put the contents of the Macintosh folder in that directory. If the folder includes other folders, they, too, will be created. If the remote computer already has directories with the names of the Macintosh folders those directories will be used.

Uploading of folders will not work with remote computers that do not support sub-directories (e.g. CMS).

The progress of the file transfer, including the name of the file being transferred, is shown in the Status panel. You can cancel the transfer by pressing the **Stop** button.

The keyboard shortcut for the **Put Folders and Files** command is Cmd-Option-T.

Under System 7.5 and later, you can upload files and folders by dragging them from the Finder into Fetch transfer windows or directory shortcuts. In that case, Fetch will use the format indicated in the transfer window.

## Quit

The **Quit** menu command closes the current connection(s) and quits the program. File transfers or other operations in progress will be cancelled.

Tip: If you happen to accidentally quit Fetch in the middle of a download, you may be able to resume the download, rather than starting from the beginning of the file. For more information about this potential time-saver, see the "Resume Download" topic.

## Raw Data

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 sent without any interpretation or conversion. Some Macintosh programs create files with all their information in the data part (e.g. *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 Raw Data format is a simple way to transfer files. However, if the file includes a resource part, that information will be lost. In any case, the file type and icon will not be preserved. So the Raw Data format should only be used if the file is strictly a data file that can be used by some other computer (e.g. an IBM PC running the PC version of *Excel*). If the file will be retrieved by a Macintosh, either MacBinary or BinHex format should be chosen.



## Rename File

This menu item allows you to change the name of a file or directory on a remote computer. If a file is currently selected, you will be prompted for the new name; otherwise you will be prompted for the current name and a new name. Try to choose a name that will be acceptable to the other computer: some systems have file name length limits and do not allow certain characters. If you specify an improper name, you will get an error message.

On Unix servers, and some others, it is possible to move a file to a new directory by renaming it. For example, renaming "myfile" to "newdir/myfile" moves it into the "newdir" directory. With System 7.5 and newer, it is similarly possible to move a file into a subdirectory by dragging the file icon to the directory icon, or into a parent directory by dragging the file icon to the "open folder" icon above the file list.

## Resume Download

This menu command allows you to resume a partially-completed download. If a download is interrupted by a network or server error, or is cancelled, it is **sometimes** possible for Fetch to resume it without re-downloading the part of the file that you already have. There are a number of limitations on this feature. Fetch can only successfully resume a transfer 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, and (if the file is a text file) implements it 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 (select Binary in the transfer window Mode pop-up), but that may give you a file with incorrect line endings.

If all of these conditions can be met, the **Resume Download** command can save a lot of time. When you choose the menu command, you will see a list of the downloads that can be resumed. You may also select files and click the **Remove** button to have Fetch forget about them (this frees up the disk space taken by the partial file). To resume a transfer, click its name and then click **OK** (or else double-click the name). You can also resume a transfer by double-clicking the partial file in the Finder. Fetch will reconnect to the appropriate server (if necessary), and attempt to complete the download.

## S/Key™

S/Key™ is a challenge-response one-time password system developed by Bellcore (S/Key is a trademark of Bellcore). 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.

Fetch offers automatic support for the S/Key system. If Fetch detects the S/Key challenge, it transparently 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. To be warned in this case, check the **Warn before sending password insecurely** checkbox in the **Security** section of **Preferences**.

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.

Fetch now also supports the One Time Password (OTP) system, which supercedes S/Key. See the "One-Time Password" topic.

## Scripting

Fetch supports a number of Apple Events. This makes it possible to automate Fetch's operations with a scripting tool such as AppleScript or UserLand Frontier. It also makes it possible for other programs to use Fetch as a "helper" application for resolving FTP Uniform Resource Locators (URLs). This help topic describes Fetch's support for the URL, Anarchie, and Fetch Apple Event Suites. In addition, Fetch supports Apple Event recording. The easiest way to find out how to script an operation is to turn on recording in your script editor, perform the operation manually, and then examine the script that has been recorded. For security reasons Fetch will not record passwords, so when you run the recorded scripts you may have to furnish passwords. Finally, Fetch supports Frontier Menu Sharing, which makes it possible for UserLand Frontier to add scripts to Fetch's menu bar.

## URL Suite

Fetch supports the **geturl** (GURL) event from the URL Apple Event Suite, 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 4.0"
  geturl "ftp://ftp.fetchsoftworks.com/"
end tell
```

The **geturl** event is used by the Internet Config system and other applications to pass URLs to other applications for resolution. So e-mail programs, newsreaders, and even web browsers can pass FTP URLs to Fetch using **geturl**.

## Fetch Suite

Fetch's Apple Event suite follows the Apple Event Object Model, which is the "organizing principle" behind AppleScript and Apple's scripting strategy. The object model can be thought of as a language where Apple Events are verbs and Apple Event Objects are nouns. The goal of this language is to be "small" but still expressive. So the number of "verbs," or events, has been kept small, but these verbs are supposed to be generic enough to work with many nouns. And the nouns (i.e. objects), while many in number, are all referred to in the same way. An object can be anything in an application: a window, a paragraph of text, a shortcut, anything that the user might want to manipulate. Under the object model, you refer to an object by first referring to the object that it is contained in. So you talk about "shortcut 3 of shortcut list window 1," or "remote file 'README' of transfer window 'ftp.fetchsoftworks.com'." The "top-level" container is the application itself. So objects are said to be "elements" of other objects, and each object class defines what kind of elements it can contain.

To make this more specific, Fetch supports the following object classes:

- 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 directory** objects
  - **shortcut list** window objects, which can contain:
    - **shortcut** objects
- the **shortcut window** object, which can contain:
  - **shortcut** objects
- **text window** objects

- the **help window** object
- the **transcript window** object
- **url** objects

Objects also have properties. For example, the application object has all the preferences settings as properties, so you can refer to its **download folder** or **default binary file type**.

Apple event objects are manipulated with events. The standard "Core" suite of events includes **close**, **count**, **delete**, **duplicate**, **exists**, **get**, **make**, **move**, **open**, **quit**, **save**, and **set**. To this list Fetch adds **download**, **view**, **view file list**, **put into**, and **send to**. Here are examples 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 beginning of alias "Hobbes:Desktop Folder:"
set fexists to exists shortcut "cool site" of shortcut list window "Fetch Shortcuts"
get use internet config
make new transfer window at beginning with properties {hostname:"ftp.fetchsoftworks.com", initial
directory:"/pub"}
move shortcut 1 of shortcut list window "Fetch Shortcuts" to beginning of shortcut list window
"Best Shortcuts"
open parent directory of transfer window 1
quit
save text window "ftp.fetchsoftworks.com messages" in alias "Hobbes:server messages"
set transfer mode of transfer window 1 to Binary
download url "ftp://ftp.fetchsoftworks.com/pub/mac/README"
view remote file "README" of transfer window 1
view 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"
```

One of the best ways to get a feel for scripting Fetch is to run your script editor, turn on recording, and then go mess around in Fetch. Most user actions will be recorded as script statements. 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.

## Anarchie Suite

Peter N Lewis's Anarchie was the first Macintosh FTP client to support scripting via Apple Events. Fetch supports most of the Anarchie suite of events, to make it easier to reuse scripts written for Anarchie (and because it is a straight-forward and functional collection of events). Descriptions of the supported events follow:

To download a file: **fetch** alias "My hard disk:my folder:cool file" **host** "ftp.domain.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.domain.com" **user** "fred" **password** "secret" **path** "uploaddir/new.file"

To delete a file or directory: **remove** **host** "ftp.domain.com" **user** "fred" **password** "secret" **path** "remotedir/bad.file"

To view a file list: **list host** "ftp.domain.com" **user** "fred" **password** "secret" **path** "remotedir"

To create a directory: **mkdir host** "ftp.domain.com" **user** "fred" **password** "secret" **path** "remotedir/newdir"

To rename a file or directory: **rename host** "ftp.domain.com" **user** "fred" **password** "secret" **path** "remotedir/old.name" **newname** "new.name"

To send a raw FTP command: **sendcommand host** "ftp.domain.com" **user** "fred" **password** "secret" **command** "SITE A LIST"

To search for a file name: **index host** "ftp.domain.com" **user** "fred" **password** "secret" **path** "remotedir/name.to.find"

The **user** and **password** parameters can be omitted if you are using anonymous FTP. 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.domain.com/remotedir/cool.file.hqx"

## Search

Some FTP servers support a command ("SITE INDEX") to search an entire FTP server for file and directory names that include a certain string. This makes it possible to find a file on an FTP server without knowing which directory it is in, and without manually navigating through all the directories. The **Search** command in the **Remote** menu prompts you for the name (or partial name) you are looking for, and creates a shortcut list with shortcuts for all of the matches. You can then open or download the matches by double-clicking (see the **Shortcuts** topic).

## Security

Fetch is now available with special security features (on PowerPC Macintoshes only; the following features are not available on 68K Macs). This help topic is intended to provide some background on the security technology used in Fetch, and describe how to use the new features.

### Encryption Basics

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, watching your password and all your 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).

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 Mac and the server. This scrambling process does take some time, and may slow down sessions depending on the speed of the Mac and the load on the server. For that reason, Fetch makes it possible to turn encryption off, in case speed is a higher priority than privacy. But, even then, passwords are exchanged in a special way that keeps them from being captured on the network.

### Kerberos

Fetch offers security features using a protocol called Kerberos. 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. 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 user ID and password can grant a user access to many services). The most widely deployed version of Kerberos at this time is Kerberos 4, and the most popular implementation of Kerberos 4 for the Macintosh is KClient, originally developed by Cornell University. The latest version of Kerberos is Kerberos 5. MIT has developed a Macintosh implementation of Kerberos 5 accessed via the Generic Security Service (GSS) programming interface.

### Using Fetch Security Features

The new security features in Fetch require special support on both ends of the session, the server and the client Macintosh running Fetch. The server must support either Kerberos 4 or Kerberos 5 (at the time of this writing, there are only a few servers with that support, and they all require some flavor of Unix). Your Macintosh must have the necessary client software, KClient 1.8 or above in the case of Kerberos 4, MIT Kerberos for Macintosh 2.0 or later for Kerberos 5. Most importantly, you must have an account in the Kerberos realm that the server uses for authentication. Having all the right software will not help you if the server does not recognize you as an authorized user.

Using Fetch with security software installed, you will notice changes in the following areas:

- **New Connection**

The "New Connection..." window has a "Security" pop-up menu to let you select the type of security to use. The default is "Cleartext password;" this is the same way Fetch has always worked, does not



require any special server support, and does not provide any protection to your password or data. The other options are "KClient," which uses the KClient Kerberos 4 software, and "GSS," which uses MIT's GSS software to work with Kerberos 5 servers. Choosing either KClient or GSS enables the "Encrypt session" 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 sent instead. It may be necessary for you to type a password later in the login sequence; if so, you will be prompted.

- **Enable Encryption**

The **Enable Encryption** menu item in the **Remote** menu lets you turn encryption on or off during an FTP session that is using either KClient or GSS security. For example, you might disable encryption for a large download that does not need to be protected.

- **Shortcuts**

Fetch shortcuts can be flagged to indicate what sort of security software to use, and whether to encrypt the session.

- **Preferences**

The "Security" section of Preferences contains options to tweak Fetch's security support, and work around common problems in secure FTP servers. The **Warn before sending password insecurely** option ensures that you will always be warned before Fetch sends your password unencrypted (e.g. because a server does not support Kerberos). The **Warn before sending cleartext password to secure server** option lets you know when you are not taking advantage of a server's security support. **Always check protection level on incoming data** is meant to catch situations where the server starts sending unencrypted data; however, at this writing, servers do this quite commonly, so it is usually more convenient to skip this check. **Make Kerberos v5 tickets forwardable** allows certain ftp servers to further authenticate you to other services which may be required such as a network file system (i.e. AFS). **Send dummy password to Kerberos servers** works around a bug in some servers available at this writing. Connecting from behind a Network Address Translation (NAT) gateway may require unchecking the **Specify GSS channel bindings** box.

## References

For more information about Kerberos, including sources of KClient and GSS software, see <<http://web.mit.edu/kerberos/www/>>.

## Credit

The security support in Fetch was written by Miro Jurisic, with the support of Marshall Vale and the rest of the MIT Mac IS team.

## Send FTP Command...

Most file transfer tasks can be handled with Fetch's regular buttons and menu items, but some file transfer servers implement non-standard commands, or require use of standard commands in non-standard situations. In these cases, you can send an arbitrary FTP command with the **Send FTP Command(s)** menu item. It prompts you for one or more commands and passes them along to the other computer without interpretation.

Note that 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 file listing commands are "NLST" and "LIST", respectively. FTP commands can be entered in upper or lower case.

If you send the "USER" command, Fetch will automatically prompt you to enter a password. If you send a file listing command ("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

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 an equivalent command called "quote." So if you have been instructed to type "quote blah blah", in Fetch you'll want to enter "blah blah" in the **Send FTP Command...** dialog box.

## Server-to-Server Transfers

Fetch 4.0 supports transferring files and directories from one server to another. If the servers in question support it, and encryption is not being used, the transfer will take place directly between the two servers. This can offer dramatic performance advantages if the connection between the two servers is much faster than your connection to one or another of them. If a direct transfer is not possible, or not safe, the data will be transferred from one server to your Macintosh, and then immediately to the other server. You can tell Fetch to not try direct transfers by unchecking the relevant option in the **Obscure Options** popup menu in the Misc. section of Preferences.

Using System 7.5 or later, to transfer a file or directory from one server to another, connect to both servers and drag the items between the file lists of the two transfer windows. You can stop the transfer by pressing the **Stop** button in either window. It is also possible to synchronize two folders (directories) using the **Mirror Folder** command in the **Remote** menu; see the "Mirror Folder" help topic for more information.

Control-clicking on a file brings up a popup menu that includes a **Duplicate** command. The **Duplicate** command is a special case of server-server transfers. Fetch opens a second connection 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 file name (e.g. myfile.1 for a duplicate of myfile) some servers will assign a meaningless name.

It is also possible to make a new copy of a file in another directory by option-dragging the file to the target directory icon. The plus sign badge on the cursor indicates that the file will be copied rather than simply moved.

## Set Permissions

Access to files on FTP servers is typically governed by permission settings that restrict which users can read, write, or execute the files. 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). Fetch provides two commands for dealing with permissions. **Set Permissions...** changes the permissions for the selected files and directories. **Set Upload Permissions...** sets the permissions to be used for files that are uploaded.

The **Set Permissions...** and **Set Upload Permissions...** dialog boxes let you specify permissions by checking boxes corresponding to read, write, and search/execute permission for an item's owner, group, and everyone else. The checkboxes are simultaneously translated into the three-digit permission numbers used by Unix, in case that format is more familiar.

To verify that permissions are set correctly, you can use the **View File List** command. If these commands do not work, it is likely that the server does not support the Unix-derived permission commands ("SITE CHMOD" for changing permissions, and "SITE UMASK" for changing upload permissions).

## Shortcuts

Shortcuts are “pointers” to files or directories that are available over the Internet. Fetch uses shortcuts in two ways. First, individual shortcuts can be saved in small files. When Fetch opens these files, it automatically opens the referenced directory, or downloads the referenced file. Second, Fetch can organize shortcuts into shortcut lists. These lists are displayed in their own windows in Fetch, and are a convenient way to organize groups of shortcuts. Shortcut lists can be saved as files and opened.

“Fetch Shortcuts” is a special shortcut list that is automatically saved (in the Preferences folder) and automatically re-opened. The shortcuts in “Fetch Shortcuts” appear in the Shortcuts pop-up menu in the **New Connection...** dialog, and the **Open Shortcut** hierarchical menu under the **File** menu. To open the “Fetch Shortcuts” shortcut list, choose **Fetch Shortcuts** from the **Window** menu. The easiest way to add a new shortcut to “Fetch Shortcuts” is to choose **Create New Shortcut...** from the **Shortcuts** pop-up menu in the **New Connection...** dialog after filling in the fields of that dialog.

To save an individual shortcut in a file, select the remote file or directory in question, and choose **Save Shortcut...** from the **File** menu. In System 7.5 and later, you can save a shortcut for the current directory by dragging the small open-folder icon to the Finder. These shortcut files can be opened by double-clicking them in the Finder or choosing **Open...** from the **File** menu.

To create a shortcut list, choose **New Shortcut List** from the **File** menu. There are a number of ways to add a shortcut to a list. The **New Shortcut...** item in the **Customize** menu lets you create a shortcut manually. You can also paste Uniform Resource Locators (URLs) into a shortcut list. In System 7.5 and later, you can drag file names or the open-folder icon from the Fetch transfer window into a shortcut list to create shortcuts for those items. You can also drag text URLs, for example from an e-mail program or newsreader, into a shortcut list.

Shortcut lists can be sorted by clicking on the column headings. In System 7.5 and later (but not on Mac OS X) you can reorder them by dragging individual shortcuts to new positions in the list (note that sorting the list results in the loss of any custom ordering you may have done). Individual shortcuts can be cut and pasted or dragged between shortcut lists. The entire lists can be saved (with **Save** from the **File** menu) and opened (with **Open...** from the **File** menu). Fetch can also open shortcut lists created by Anarchie.

To open a directory that is referenced by a shortcut in a shortcut list, simply double-click the shortcut or press Enter while it is selected. Likewise, double-clicking or pressing Enter will prompt Fetch to download a file referenced by a shortcut. If you are using System 7.5 or later, you can also drag the shortcut to a Finder window or the desktop in order to initiate the download. Similarly, uploads can be initiated by dragging files and/or folders to a shortcut for the destination; Fetch will automatically open a connection and perform the operation.

To edit a shortcut in a shortcut list, first open the shortcut list window (e.g. by choosing **Fetch Shortcuts** from the **Window** menu), click on the shortcut to select it and choose **Edit Shortcut...** from the **Customize** menu, or Option-double-click on it. Control-double-clicking on a shortcut in a shortcut list brings up the **New Connection...** dialog, filled in with information from the shortcut.

NOTE: If you save a password as part of a shortcut, that password will be no more secure than the disk it is saved on. Be certain that others do not have access to your disk before saving a shortcut that includes a password. In Mac OS 9 or later, you can instead save passwords in the Keychain, where they are protected by your keychain password.

## Status Panel

The area on the right side of the Fetch window is the Status panel. It contains information about what Fetch is doing at any given time. If a file transfer is in progress, it displays the name, format, and approximate size of the file being transferred, the amount of data that has been transferred so far, the average speed of the transfer so far, and an estimate of how long it will take for the transfer to complete. If any of these fields is missing, it is because the information is unavailable. If the window is large enough, there is also a pie- or doughnut-shaped progress graph, indicating how much of the file has been retrieved. If an operation is in progress, there's a **Stop** button at the bottom of the status panel.

## Stop

If a time-consuming operation is in progress, the **Stop** button will appear at the bottom of the status pane, on the right side of the window. You can stop the current operation by pressing this button or pressing Cmd-. Once you have done so, the button will disappear and the Status line will read "Cancelling...."

Most computers support the stop command; with these machines, it will only take a few seconds. Some machines, however, ignore the command and insist on completing the current operation. In these cases, the status line will read "Cancelling..." for quite a while. The only ways to quickly exit from this situation are to close the connection or quit from Fetch.

## Suffix Mapping

When the **Automatic** button is selected, Fetch looks for standard filename suffixes in order to decide whether a file is a text file or a binary file and, if it is a raw binary file, what type and creator to assign to it.

If your system has the Internet control panel (System 8.5 through 9.x) or if you have an older system with Internet Config installed, and have checked the **Use Internet Config** checkbox in **Preferences** (see the "Preferences" topic), Fetch uses the suffix mappings maintained by that control panel (see the **Internet Config** topic); open it to change the settings.

Otherwise, you can modify the list of recognized suffixes in the **Suffix Mapping** dialog.

A suffix mapping associates a suffix (such as ".hqx") with a file type (either Text, Binary, or BinHex) and, optionally, with type/creator codes. The "Text" type means that files with the associated suffix should be downloaded as text files; similarly, the "Binary" type means that the files should be downloaded as binary files. The "BinHex" type means that the file should be downloaded as text (since BinHex files are a kind of text file) but that non-BinHex material should be ignored. If a type/creator pair has been specified, text or raw binary files will be given that type and creator. For instance, a file with the ".doc" suffix is usually given the type/creator pair of W8BN/MSWD (the codes used by *Microsoft Word*), since ".doc" is the standard ending for Microsoft Word files.

Suffix mappings are only used to distinguish text and binary downloads when the Automatic download mode is selected (they may be used to assign type/creator codes even when Text or Binary is selected). Information in the file overrides the application-specified in the suffix mapping: if a file is in BinHex, MacBinary or AppleSingle format, it will include information about the application it belongs to.

Each new release of Fetch includes new default suffix mappings. To use these rather than your current settings, quit Fetch and delete the Fetch Prefs file in the Preferences folder. Note that you will lose any preference settings that you have changed from their default values.



## TCP/IP

### Open Transport

If you're using Mac OS 7.6 or newer, the system uses Apple's Open Transport implementation of the TCP/IP networking protocols; it was introduced in July 1995.

### MacTCP

MacTCP was Apple's original implementation of TCP/IP. MacTCP, originally a commercial product, was included with System 7.5 and was last supported in System 7.5.3.

System software 7.5 through 7.5.3 shipped with both Open Transport and MacTCP and a Network Software Selector control panel to switch between the two.

Once a network connection has been established, a message in the About Fetch... dialog box indicates whether Open Transport or MacTCP is being used.

## Uniform Resource Locators

Uniform Resource Locators are global names for items on the Internet: they specify what kind of resource the item is, what server it can be found at, and its name on that server. In the case of files and directories on FTP servers, a URL is a string in the following format:

```
ftp://<userid>:<password>@<hostname>:<port>/<path>
```

If the userid and password are omitted it's assumed to be an anonymous FTP server. The port value is also optional. So the URL for a copy of Fetch on Fetch Softworks' FTP server would look like this:

```
ftp://ftp.fetchsoftworks.com/Fetch.hqx
```

URLs were originally designed for use in the World Wide Web (WWW). Fetch uses URLs as part of shortcut files, and can also use URLs in the **New Connection**, **Edit Shortcut**, **Change Directory**, **View Text File**, **View Media File**, **Delete Directory or File**, **Get Info**, **Get File**, and **Get Directories and Files** dialogs. The **Copy** command in the **Edit** menu puts a URL for the current directory or selected items on the clipboard. Dragging the open-folder icon to a text editor inserts a URL for the current directory. Similarly, you can create shortcuts in a shortcut list by pasting or dragging a URL into the window.

Most Macintosh Internet programs can be used as URL "helper" applications. Programs can pass URLs to the appropriate helper application, which can retrieve the desired data or perform the desired action. Fetch can be used as a helper for FTP URLs (i.e. ones that begin "ftp://").

Fetch also supports a user interface convention for more convenient handling of URLs. Command-clicking on a URL in any of Fetch's text windows (such as this help window) automatically passes that URL to the helper program you have specified using Internet Config (see the **Internet Config** topic). So, for instance, command-clicking on <http://fetchsoftworks.com/fetch.html> will instruct your web browser to view the official Fetch web page.

## View File List

Use this command to view the file list for the current directory in a separate text window. This listing will often include information (such as file permissions) that Fetch does not display in its own file list. You can search for information in text windows with the **Find** command, print them, and save them as text files. The font and tab width can be changed with the **Set Font and Size** command.

## View File

Fetch has two commands for viewing files on an FTP server without leaving Fetch: **View Text File** and **View Media File**. **View Text File** is used, not surprisingly, for text files (including HTML files). **View Media File** is used to view images, sounds, movies, animations, or any other sort of file that can be displayed by Apple's QuickTime software.

In either case, Fetch will download the selected file and display it in a separate window.

## Troubleshooting

*When I upload GIF or JPEG files in MacBinary format, my web browser can't display them.*

Web browsers (and many other programs) do not expect to find files encoded in a format like MacBinary III; they expect to find the raw data. Use the Automatic or Raw Data format for uploading web graphics, or other cross-platform files.

*When I try to open a connection, I get a -3221 or a -3264 error.*

Those are Open Transport errors, and usually indicate a problem with the configuration of your TCP/IP software, or with your connection to the Internet. Check the TCP/IP (or MacTCP) settings with your Internet Service Provider's tech support department.

*Fetch can download files, but when I try to upload a file over my modem connection, the computer freezes, or the upload fails.*

There is probably a problem with the flow-control settings of your dial-up software or your modem. Make sure that your modem init string enables hardware handshaking, and that your dial-up software is configured to do hardware handshaking. Consult with your modem manual or service provider for more information.

*When I try to open a connection, my computer freezes.*

Check the configuration of your SLIP or PPP software, and consult with the administrators of the dial-up service you are using. This sort of problem is usually caused by misconfiguration of the dial-up software.

*When I try to get a file from an IBM MVS server, Fetch creates an empty folder.*

This is due to a bug in the server (and affects some other servers besides IBM's). The workaround is to check the **Treat mystery items as files** checkbox in the **Misc.** pane of **Preferences**.

*When I transfer a file over my dial-up link, Fetch slows down and eventually drops the connection.*

Check the configuration of your SLIP or PPP software, and consult with the administrators of the dial-up service you are using. This sort of problem is usually caused by misconfiguration of the dial-up software.

*The file list includes other information besides the file name, and the Get File button doesn't work.*

Select the **Refresh File List** command in the **Directories** menu; that will usually fix this problem. Otherwise, if you know the name of the file, you can hold down the Option key when you click the **Get File** button and type in the file's name.

*The file list does not include a file that I know was just put there.*

Select the **Refresh File List** command in the **Directories** menu; that will usually fix this problem. Fetch reuses file lists when they have not changed in a long while, but this makes it possible to miss new changes.

*When I tell Fetch to send the FTP command "ls", I get an error message about the command "LS"; why does Fetch capitalize the command?*

Fetch does not capitalize the command, the server does. In fact, capitalization of FTP commands does not matter. The problem is that "ls" is not an FTP command; it is a command used in many command-line FTP clients, such as the standard Unix client, but it is not a part of the FTP protocol. The corresponding FTP command is "NLST". Similarly, "dir", "get", "put", and "cd" are not FTP commands; the proper commands are "LIST", "RETR", "STOR", and "CWD", respectively.

*The file names in the file list don't look right -- there is size and date information in the name column.*

Fetch sometimes guesses what kind of computer it is talking to, and it can guess incorrectly. You should be able to correct the problem by refreshing the file list -- choose the current directory from the directory pop-up menu above the file list.

*The downloaded files have the wrong icon.*

This problem can be caused by a number of factors. First, be sure that you are using the right settings -- if you've tried the **Automatic** setting, try **Binary**. It may also be helpful to read the "Automatic/Text/Binary" help topic under "Reference."

*I can connect to a machine on my local network, but not to Internet FTP servers.*

Double-check the Router address in the TCP/IP control panel or, on older systems, the Gateway Address of the MacTCP control panel.

*I get the message "This is not a proper BinHex format file."*

This means that Fetch could not successfully de-binhex the file. Deselect the **BinHex4** checkbox in the **Download** pane of **Preferences** and download again (see the "Preferences" help topic under "Reference"). Then try to de-binhex the file with StuffIt Expander (available from [www.aladdinsys.com](http://www.aladdinsys.com)).

*I get the message "This is not a proper MacBinary III file."*

This message, and another one about AppleSingle format files, means that the file has been corrupted in some way. First, try downloading again in binary format. Next, contact the source of the file to get a new copy. If the file was created by Fetch, send a report to [support@fetchsoftworks.com](mailto:support@fetchsoftworks.com) so we can try to find the bug.

*The Stop button doesn't work when I'm connected to a VMS Vax.*

We have observed problems with Stop on VMS and CMS systems. We're working on these problems; in the meantime, you can cancel a transfer by closing the connection altogether.

## Glossary of Terms

*anonymous FTP* -- the process of connecting to a remote computer that does not require any identification or password. In this case the user name used is either blank or "anonymous" and the password is optional. Several archives of Macintosh software are available to all by anonymous FTP.

*binary file* -- a file that may contain any characters in any format; not a text file. Files in MacBinary III 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.

*directory* -- a collection of files and sub-directories, equivalent to Macintosh folders. Sub-directories are directories within some other directory; the enclosing directory is known as the parent directory. Most systems have a notion of the current directory, the directory whose files you are viewing at that time. Not all systems have directories, and some systems do not permit sub-directories.

*download* -- the operation of moving a file from some other computer to your own.

*host* -- the remote computer you are connected to. This computer runs a program that accepts your connections, and therefore can be seen as your host. Most host computers are specialized servers, but a personal computer can also act as a host.

*Internet* -- a collection of networks linking schools, universities, companies, and individuals around the world. FTP is the standard file transfer protocol for the Internet.

*IP number* -- a type of computer address of the form number.number.number.number (e.g. 128.170.16.4). Usually you will refer to machines by their names (e.g. "ftp.fetchsoftworks.com") but Fetch can also connect to machines with just their numeric address.

*mirror* -- as a verb, the operation of synchronizing files between two computers by transferring the ones that have changed. Mirroring is an efficient way to upload a web site, since only new and changed files are transferred.

*Point-to-Point Protocol (PPP)* -- a protocol for using network protocols, including TCP/IP, over a serial line (e.g. a dial-up modem connection). A Macintosh with PPP software (such as MacPPP) and a connection to a PPP service provider can run Fetch.

*proxy server* -- a special FTP server that acts as an intermediary between an FTP client and the FTP server that the client really wants to connect to. Proxy servers are used on networks where most machines are not permitted to make Internet connections; instead those machines must connect to the proxy, which in turn connects to the desired server. Fetch supports a number of proxy servers (see the **Preferences** topic). Web proxy servers, which use the HTTP protocol rather than FTP, are not supported.

*text file* -- a file containing lines of letters, numbers, and punctuation. Most word processors can create text files and some editors (such as *SimpleText* and *BBEdit*) only create text files. A non-text file is a binary 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, the protocol used by Fetch, is part of the TCP/IP family of protocols. Fetch uses TCP/IP services provided by Apple's MacTCP or Open Transport software.

*Uniform Resource Locator (URL)* -- a global name for a resource available on the Internet. One example of a URL is "ftp://ftp.fetchsoftworks.com/example/Fetch\_4.0.sit.hqx." In this case the resource is a file available for FTP on host ftp.fetchsoftworks.com, with the full name /example/ Fetch\_4.0.sit.hqx. You can use URLs to tell Fetch what file or directory to access. You can get the URL for the currently selected file or directory in Fetch by choosing the Copy command in the Edit menu. URLs were developed as part of the World Wide Web (WWW) system on the Internet.

*upload* -- the operation of moving a file from your computer to some other one.



## Release Notes

Fetch 4.0.3 has the following new features and bug fixes since Fetch 4.0.2:

- Fixed AppleScript recording on OS X 10.2.
- Fixed a bug that caused crashes when canceling transfers on OS X 10.2
- Added a preference to specify the text file line ending style.
- Fixed dozens of other bugs.

Fetch 4.0.2 has the following new features and bug fixes (since Fetch 4.0 and 4.0.1):

- Drag and drop compatibility with Mac OS X 10.1.3.
- Fixed bugs that required setting passive mode or resulted in -3155 errors on Mac OS X.
- Fixed bugs that kept firewall settings from being remembered on Mac OS X.
- Fixed bugs that kept the Edit with BBEdit and Edit with GraphicConverter features from working on Mac OS X.
- Added support for longer-than-31-character file names on Mac OS X.
- Improved the mirror folder feature for servers that do not support the MDTM command.
- Reduced CPU usage on Mac OS X.
- Improved feedback for drag-and-drop operations.
- Shared menu support on Mac OS X.
- Added the ability to duplicate remote files.
- Fixed dozens of other bugs.

Fetch 4.0 has the following new features and bug fixes:

- Appearance Manager support (Mac OS 8.5 and later), including color icons, resizable file list columns, selectable file list font & size, and translucent icon dragging.
- Mirror command to duplicate a local or remote folder with a remote or local folder; mirror settings can be saved in mirror documents for one-click website updating.
- Server-to-server transfers. Drag items between connection windows; they will be transferred directly if possible, indirectly otherwise. The Mirror feature supports server-to-server mirroring.
- Edit File with BBEdit and Edit File with GraphicConverter allow automatic down/uploading of files to be edited. Also, "Edit New File with BBEdit/GraphicConverter", which lets you create new remote files from within Fetch.
- Keychain support (Mac OS 9 and OS X).
- Support for KClient (Kerberos 4) and GSS (including Kerberos 5) security.
- Recent Connections list (in the File menu).
- View Media File command to preview images, sounds, movies.
- Contextual menus -- Control-click on remote items or shortcuts to see a list of relevant commands.
- Resume downloads even after Fetch quits or crashes.
- Move files and folders by dragging the icons around (only works on Unix and Unix-like servers).
- Get Info command to total size of folders.
- View by Kind.
- New transfer window Mode and Format pop-up menus for setting download and upload options, and new window layout.
- Better compatibility with Mac FTP servers (e.g. AppleShare/IP, Rumpus).
- Cmd-delete to delete.
- Set the modification date of downloaded text and binary files to whatever it was on the server.
- Create new shortcuts from the New Connection window.
- Drag-upload to directory icons (not just current directory).
- Notification options, including a custom bark sound.
- Automatic format for uploads.

- Optionally add suffixes to uploaded files.
- MacBinary III support.
- Support for 11 different kinds of proxy servers.
- Support for retrying connection attempts.
- Fixed a bug in S/Key support.
- Fixed a bug with file names with leading spaces.
- Removed obsolete features.
- A number of other minor bugs were fixed.

## Credits

Fetch was written by Jim Matthews. This copy of Fetch is a product of Fetch Softworks, under license from Dartmouth College.

Thanks to:

- Richard Nelson, for the icon and "running-dog" animation.
- Punch Taylor, for supporting Fetch's development from 1989 on.
- Tadashi Suzuki and Hart Computer, for the new logo.
- Doug Grinbergs, for finding dozens and dozens of bugs.
- All the users who reported problems and suggested new features.
- All the individuals and companies who licensed version 3.0.
- Cathy MacDonald and Chapin, for the bark.
- ABC's "Who Wants To Be A Millionaire," for the seed money.
- Margie, for her love, faith, judgement, endurance, and order processing.
- Dylan, for being my defense minister and technology adviser.
- And to Daniel, for being Daniel.