UNIVERSITY OF
CAMBRIDGE

Computing Service

G72

File Transfer Using SCP, SFTP or FTP

many leaflets can be found at:

`http://www.cam.ac.uk/cs/docs`

---

50 pence
April 2003

File transfer using SCP, SFTP or FTP

Contents

Introduction

This leaflet describes methods of transferring files from one Internet-connected system to another, using Secure Copy (SCP), File Transfer Protocol (FTP), or Secure File Transfer Protocol (SFTP).

Before using any of these programs, you should check the notes in Computing Service Information Sheet 42: *Transferring files between computer systems*, about (a) making sure the file will be usable at the other end of the transfer and (b) alternative ways of transferring files.

Servers and clients

To transfer files by any of these methods, you need to run a *client program* on the system which initiates the connection, and the system at the other end (the *host*) needs to be running a *server program* which can accept incoming connections and process requests from your client. Once connected, you can cause files to be transferred in either direction (host to client or client to host).

Generally, large multi-user systems can be expected to accept incoming FTP connections, and many will also run an *SSH server* which accepts SCP or SFTP connections. It may be possible to run the same servers on your own machine but this is not necessary for most users, who only need to run client programs. There are major security issues to be taken into account if you do want to run your own server.

If you are transferring files to or from your own filespace on the host system, you will normally be asked for a password. Where files are made publicly available for *anonymous FTP*, it is usual to provide your e-mail address in place of a password.

Obviously you can only copy files which you have permission to read on the source system, and can only copy them to filenames which you have permission to create on the destination system.

Comparison of FTP, SCP and SFTP

FTP or File Transfer Protocol has been in use for many years as the primary method for transferring files from one computer system to another. FTP clients and servers are very widely available. However, there are security problems with FTP; when you set up an FTP connection, your password on the remote system is transmitted without any form of encryption. FTP is therefore not generally recommended, except for anonymous FTP.

Furthermore, there are security issues associated with some FTP servers, which have led to a block being imposed on incoming FTP transfers to Cambridge systems, except for a list of approved systems (see Appendix 1 for more details).

The secure alternatives to FTP are SFTP (rather like FTP, with slightly fewer facilities but with extra security features) and SCP (Secure Copy). Neither of these is universally available, and they have different advantages.

The main difference in practical operation between SCP and SFTP/FTP is that for SFTP or FTP you set up a connection and can then carry out a whole series of operations (including many which are not actually transfers, such as listing or even deleting files on the remote system). With many SCP clients, a single command normally transfers a single file or group of files, and to list files on the remote system, for instance, you have to login as a separate exercise (normally using ssh, the secure alternative to telnet). Note that WinSCP, although described as an SCP client, provides facilities more like those of FTP or SFTP.

Computing Service systems

All public Computing Service systems (the PWF, Hermes, CUS and Pelican) run FTP servers. CUS, Hermes and Pelican also run SSH servers for SFTP and SCP, but the PWF does not. Hermes cannot, however, accept connections from WinSCP.

At the other end of the transaction, CUS and all PWF systems provide client programs for FTP and for SCP. SFTP clients are available on CUS and on PWF Linux and PWF Macintoshes, but not on the PWF Windows machines.

Client programs are not provided on Hermes and Pelican since these systems do not allow users to run general programs.

System	FTP client	SCP client	SFTP client	FTP server	SSH server (for SCP/SFTP)
PWF Windows	FTP Explorer	WinSCP or pscp	no	))yes	no
PWF Linux	ftp	scp	sftp	)	
PWF Mac	Fetch	NiftyTelnet	MacSFTP	)	
CUS	ftp	scp	sftp	yes	yes
Hermes	-	-	-	yes	yes
Pelican	-	-	-	yes	yes

This means that you can make transfers between any two of the above Computing Service systems; you will sometimes have a choice about which end should be the client and which the server, but not always. For transfers where the server is running on the PWF you do not have the option of SCP or SFTP.

Other computer systems

To find out whether a system to or from which you need to transfer files runs an FTP or SSH server, you need to ask the administrators of the system (or simply try one of the client programs and see if it works). Individuals running their own systems should not normally need to run FTP or SSH servers, and should be aware of security issues if they do so.

To find out about client programs, you may also need to ask the administrators of the local systems. Appendix 2 gives pointers to sources for various FTP/SFTP/SCP clients.

SFTP clients

SFTP can be used for transfers between two systems on which you have accounts (e.g. from CUS to the Pelican archiving system). Where available, it is preferable to FTP for security reasons. The other secure option, SCP, offers different features and in some cases handles error messages better; see a later section of this leaflet. The PWF does not support incoming SCP or SFTP connections.

SFTP cannot be used for *anonymous FTP (*transfers from a publicly available system not requiring a user identifier or password).

Using sftp on a Unix system

SFTP on Unix systems looks rather similar to FTP, so much of this section also applies if you are using FTP. SFTP does not have certain of the FTP facilities but many of these are provided by the `ssh` command instead, as explained later.

To invoke SFTP on a Unix system, use the command

`sftp` *`hostname`*

where *hostname* is the name of the remote system you wish to contact, e.g.

```
sftp hermes.cam.ac.uk
```

You can also start the program by typing simply `sftp`, and then use the `open` command to set up the connection you require.

You will then be asked for your password. The program reports that a connection has been made (or gives details if it fails) and then awaits further commands. The commands available vary between systems but the simple ones are common to all. On any system, the `help` command lists the SFTP commands available. The `quit` command closes the session.

Many remote systems will close the connection if it is idle for a certain time. You can re-open it using the command `open` *`hostname`*.

Finding files

Initially your home directory is used as the current working directory.

Useful commands available in most versions (use `help` to check) for navigating around the remote filing system include:

`ls`	lists the files in the current working directory
`dir`	on some systems gives a more detailed listing
`cd` *`name`*	changes to the subdirectory *`name`* within the current working directory
`cd ..`	changes to the parent directory of the current directory
`pwd`	displays the name of the current working directory

Many versions of SFTP provide similar commands for navigating through the directories on your local system (e.g. `lcd` to change directory on the local system). Others simply provide a command (usually the ! character) which escapes to the local operating system and allows you to issue whatever local commands are needed; `exit` returns you to the program.

Transferring files

The most useful commands for transferring files are `get` (remote to local) and `put` (local to remote). The syntax is

`get` *`file`*

which assumes the file is to be given the same name on the destination system as it has on the source, or

`get` *`file1 file2`*

where *`file1`* is the source name, and *`file2`* the destination name. The syntax for `put` is similar.

Example

To fetch a file called `fred` from your filespace on Hermes, and store it under a new name in subdirectory `info` on the client Unix system (e.g. PWF Linux)

```
sftp hermes.cam.ac.uk            set up a connection to the server

Password for user sjc1000:  *******

sftp> ls                         list the contents of the directory
sftp> lcd info                   point to the destination directory on the PWF
sftp> get fred newfred           fetch and rename the file
sftp> quit                       close the connection and quit SFTP
```

Other useful commands for file transfer (not available in all versions of these programs) are `mget` and `mput`, which allow multiple transfers, e.g.

`mget data*`

executes a `get` command for every file in the current directory whose name begins with `data`. Files cannot be renamed using `mget` or `mput`.

Usually you can interrupt a transfer using `ctrl/C`

Other file operations

Various SFTP programs provide some or all of the following:

`rm`	deletes a file on the remote system
`mkdir`	creates a directory on the remote system
`rmdir`	deletes a directory on the remote system
`chmod`	changes permissions on the remote system

Note that the sftp commands `chgrp`, `chown`, `ln` and `symlink` (not described here) are deliberately disabled when the host is Pelican.

Unix `sftp` does not generally provide for checking your quota, or for displaying a file without fetching its contents to your filespace. These can be done using the `ssh` command, for example

```
ssh pelican.cam.ac.uk quota -v
ssh cus.cam.ac.uk cat /info/new |more
```

Some implementations of `sftp` require a numeric argument (rather than the more usual alphabetic codes) to the `chmod` command for changing access permissions; again, as an alternative you can use `ssh`:

```
ssh hermes.cam.ac.uk chmod -or '*.doc'
```

In each case you will be asked for your password on the remote system.

Note that the SFTP protocol provides only a very simple mechanism for reporting errors. This means that, if you run out of quota while uploading files to Pelican via sftp, you will be told only that the operation has failed, but not the reason (you can check your quota using `ssh` as described above). In general the common `sftp` clients do not have good error handling facilities. For this reason, non-interactive use of `sftp` is not recommended (and is not described here). The `scp` command has better error messages and more reliable exit codes, and is thus safer for batch use.

Using MacSFTP (Macintosh)

MacSFTP is the recommended SFTP program on the Macintosh, and is available on the PWF.

Start MacSFTP by double-clicking on the program icon as usual. You should then see a dialog box for filling in details of the system to connect to (if this dialog box does not appear, select **New Connection** from the **File** menu). Fill in the hostname (e.g. `cus.cam.ac.uk`), the appropriate user identifier (in the Login box) and password, and a directory name (path) if you know which directory you need (otherwise your home directory will be taken as the starting point).

Then click on **OK**, and the connection should be made.

On your own Macintosh, you may want to store the connections you use most often as *bookmarks*; select **New Bookmark** from the **Bookmarks** menu. Fill in the details and click **OK**; you will then be able to open the connection at any time by selecting it from the **Bookmarks** menu. If you include a host password in any bookmark specification, then you will be invited to protect this information by choosing a "passphrase" which you will then have to quote in order to access any of your bookmarks that contain passwords. If you do not include the host password in a bookmark specification, you will be asked for it when you try to make the connection.

Once you are connected to your remote host, a list of files will appear. You can look at files in other directories, by double-clicking on a directory name or by using the **Remote** menu and selecting **Change Directory**.

To fetch any file, you can drag and drop the file between the remote host window and any open folder window on your desktop. Alternatively, select the file or files to be transferred and then select **Get Files/Directories** from the **Remote** menu. If you are transferring one file only, you will be offered the usual dialog box to name the Macintosh file. If you are transferring several files, you will be asked which folder to put them in, but you do not have the option of changing the names.

To send files from the Macintosh to the remote host, you can drag and drop as above. Alternatively, select **Put Files/Directories** from the **Remote** menu. The usual dialog box is provided to select the file to be sent, and it will be put in the currently open directory on the host system.

The **Remote** menu in MacSFTP allows you to delete or rename files or directories on the remote system, to change the access permissions on files, and to view a remote file without explicitly fetching it. You can also edit remote files using BBedit or Graphic Converter.

There is no facility to check your quota on the remote system using MacSFTP.

SCP clients

SCP clients are available on most Unix systems including CUS and PWF Linux, and as part of the NiftyTelnet program on a Macintosh. On the PWF PCs a graphical SCP client, WinSCP, is available and there is also a basic line-mode client pscp.

The basic SCP command carries out a single file transfer (unlike SFTP or FTP where, once connected, you can carry out any number of transfers). More complicated operations can be carried out using the `ssh` command, as described below.

Using scp on a Unix system

The simplest type of `scp` call carries out a single file transfer.

If you are using PWF Linux, for instance, to transfer a file called `test` from the PWF to Hermes without changing its name, the basic command is

```
scp test hermes:
```

You will be asked to supply your Hermes password before the transfer is made.

To fetch the same file back and change its name to test1, use

```
scp hermes:test test1
```

You can transfer a whole directory using the -r option:

```
scp -r mydir hermes:
```

Users familiar with the ssh protocols will be able to avoid the password prompts by using pure RSA authentication. The RSA public keys can be put in the file `~/.ssh/authorized_keys` in the ordinary way. Users of RSA authentication must take great care not to expose their private keys.

The `scp` command only copies files or directories; it does not easily allow you to list your files on the remote host, or delete, copy or rename them. To do any of these interactively, you need to login to the host (typically using the `ssh` command) and use whatever file management commands are available to you at the host end. Hermes, for instance, makes file management operations available through its menu system, and Pelican through the interactive file management system. Other Unix hosts such as CUS give you access to the full range of Unix file commands.

You can also add commands to the `ssh` line itself rather than logging in, e.g., you can list your Hermes files using

```
ssh hermes ls
```

or delete a CUS file using

```
ssh cus rm myfile
```

To list your Word documents stored on Pelican, use

```
ssh pelican ls '*.doc'
```

(note that the name `*.doc` is in quotation marks so that the * will not be expanded on the client machine, but passed unchanged to Pelican for expansion there.)

To find your quota on Pelican, use

```
ssh pelican quota -v
```

You can put several commands in a file, and obey them in sequence, using

```
ssh hermes <myfile
```

It is also possible to carry out file copying non-interactively using `ssh` rather than `scp`, so users who are confident with various aspects of Unix can use the `cat` command and its variants to combine packing and compressing files with sending the result to Pelican, or the reverse process. For example, the following command issued on a client Unix machine will create, on Pelican, a compressed tar archive of a directory on the client machine:

```
(cd mydir && tar cvf - .) | compress | ssh pelican.cam.ac.uk cat2 mydir.tar.Z
```

and the following command issued on a client Unix machine will fetch and unpack a compressed tar archive:

```
ssh pelican.cam.ac.uk cat mydir.tar.Z | zcat | (cd mydir && tar xvf -)
```

Using WinSCP

WinSCP is a Windows interface to SCP available on the PWF PCs (see Appendix 2 for obtaining your own copy). Note that it does not work with Hermes.

Start WinSCP from the Start menu (click on **Start**, point to **PWF Programs and Information**, **Communications and Networking**, then click on **WinSCP**). You should then see a box listing "stored sessions" of which initially the only one is `cus.cam.ac.uk`. Either select this and click Load, or to connect to any other host click on the Basic tab. Fill in the hostname if needed, and your username and password on the remote system. Then click on **Login**.

Once the connection has been made, two panels are displayed: one for your local system and one for the remote system. Each panel includes a pull-down menu for switching between directories or devices, and a button for moving up to the parent directory. You can click on any folder within the panel to open it.

You can drag and drop files between the two panels (you may not see something you have just moved in the listing until you select Refresh from the Local or Remote menu as appropriate).

Alternatively, the buttons listed along the bottom of the panel allow you to move the selected file or files from one system to the other, and also to carry out many of the operations (such as deleting and renaming files and creating directories on the remote system) not available in other SCP clients.

WinSCP provides a choice of interfaces; one as above, which is selected by default on the PWF, and the other looking like Windows Explorer. The choice between interfaces is made via the Preferences item on the Options menu.

Using pscp (PC)

This is at present the only secure file transfer program available on the PWF PCs that allows transfers to and from Hermes. It is a line-mode program and looks very much like the Unix `scp` command (see earlier section), so the details are not repeated here.

To launch pscp, you need to call up a Command Prompt window (click on **Start**, move to **Programs**, then **Accessories**, and click on **Command prompt**). The command syntax is then roughly as for Unix scp, except that by default you have to give your user identifier before the hostname, e.g.

```
pscp filename spqr1@pelican:newfilename
```

It may be useful to know that in Windows 2000, if you drag a file into a command prompt box, the name of the file will be added to the current command.

Using SCP within NiftyTelnet on a Macintosh

You need to be running at least NiftyTelnet 1.1 SSH r3 (which is the version on the PWF), and if you are setting it up yourself you are recommended to set the protocol for your connection to be SSH-3DES.

To copy a file from your Macintosh to a remote host (e.g. Hermes):

Open the Nifty Telnet program by double-clicking as usual.

Go to **File** on the menu bar and select **New Connection**.

Under the New Connection window, select **hermes** (see below for other hosts).

Now click on the **SCP** button, and the SCP window will open up.

Select **Send files/folders** and click on the **Add Files/Folders** button to choose files and/or folders you wish to copy to the host.

Under **Destination Path** enter where you want your files to go, e.g. `~myname/dir1/` Click on **Start Copy** or press the return key. You will be prompted for your Hermes username and password; if these are accepted the copying takes place.

If you are copying to or from a remote host which is not in the list of hosts presented, click on **New**, fill in a shortcut name and the full name of your host system as the Host Name, and make sure SSH-3DES is selected in the protocol box. Then click **OK** and select your new shortcut name from the list. Continue by clicking the **SCP** button as above.

To copy a file from a remote host to your Macintosh:

Follow the first four steps above.

When the SCP window opens, select **Receive Files/Folders**.

Under **Source File** enter the path for the remote file/folder you wish to copy, e.g. `~myname/dir1/somefile`

Under **Download Folder** select where you want to put the copy on your Macintosh. Click on **Start Copy** or press the return key. You will be prompted for your remote host username and password. If these are accepted the copying takes place.

FTP clients

These programs are mainly needed for anonymous FTP, but may also be required where a host (e.g. the PWF) does not support SFTP or SCP transfers.

Anonymous FTP is discussed in a later section; as far as the use of the programs are concerned there is very little difference, except that anonymous FTP has special conventions for logging in.

ASCII and binary transfers

In FTP (unlike SFTP or SCP) you have to decide whether your transfers are to be in ASCII or binary mode (in SFTP or SCP all transfers are in binary). The default mode of FTP transfer was originally designed for the transfer of pure text files between systems, and early FTP clients assumed by default that the file to be transferred was in ASCII. Some clients now default to binary transfer, as described below, but all clients give a choice between the two modes.

Many files such as compressed or tar files (see below), as well as formatted files produced by various application programs such as Word, need to be transferred in binary mode. FTP Explorer and Fetch will attempt by default to choose the transfer mode according to the type of file, but they may not always get this right; both programs provide a facility to force transfers to be binary if necessary. If you use FTP on a Unix system you may have to request binary transfer explicitly, using the command `binary`. All transfers within the current session will then be binary until you change back using the command `ascii`.

FTP via a Web browser

On any of the standard platforms (Unix, Windows, Macintosh) you can connect to an FTP server using a Web browser. To do this, enter a URL starting with `ftp:` instead of `http:`, and include your user identifier before the name of the host, for instance

```
ftp://fjc1000@pelican.cam.ac.uk
```

You will be asked for your password for the remote system, and then taken to a directory listing, which (depending on the system) may be your home directory or may be the root directory of the whole system. You can navigate by clicking on directory names until you reach the file or directory you want. To fetch a file you can double-click on it. On some systems you can also transfer a file in either direction by dragging and dropping it, but not all browsers support this feature properly (so that some of them do not offer any way to transfer a file to the host end).

This mode of using FTP does not allow you to do anything other than listing files and transferring them. It is however useful for anonymous FTP, as described later.

Using ftp on a Unix system

To invoke FTP on a Unix system, use the command

`ftp` *`hostname`*

where *hostname* is the name of the remote system you wish to contact, e.g.

```
ftp home.pwf.cam.ac.uk
```

You will then be asked for your user identifier and password (see later section for anonymous FTP).

The commands available vary between systems but the simple ones are common to all. On any system, the `help` command lists the FTP commands available. The `quit` command closes the session. Many remote systems will close the connection if it is idle for a certain time; you can re-open it using the command `open` *`hostname`*.

Finding files

Initially your home directory is used as the current working directory.

Useful commands available in most versions (use `help` to check) for navigating around the remote filing system include the following:

`ls`	lists the files in the current working directory
`dir`	on some systems gives a more detailed listing
`cd` *`name`*	changes to the subdirectory *`name`* within the current working directory
`cd ..`	changes to the parent directory of the current directory
`pwd`	displays the name of the current working directory

Many versions of FTP provide similar commands for navigating through the directories on your local system (e.g. `lcd` to change directory on the local system). Others simply provide a command (usually the ! character) which escapes to the local operating system and allows you to issue whatever local commands are needed; `exit` returns you to the program.

Transferring files

The most useful commands for transferring files are `get` (remote to local) and `put` (local to remote). The syntax is

`get` *`file`*

which assumes the file is to be given the same name on the destination system as it has on the source, or

`get` *`file1 file2`*

where *`file1`* is the source name, and *`file2`* the destination name. The syntax for `put` is similar.

Example

To fetch a file called `fred` from your filespace on the PWF, and store it under a new

name in subdirectory `info` on the client Unix system (e.g. CUS)

```
ftp home.pwf.cam.ac.uk          set up a connection to the FTP server
Name (home.pwf.cam.ac.uk:sjc1000):
Password for user sjc1000:  ******
sftp> ls                        list the contents of the directory
sftp> lcd info                  point to the destination directory on CUS
sftp> get fred newfred          fetch and rename the file
sftp> quit                      close the connection and quit FTP
```

Other useful commands for file transfer (not available in all versions of these programs) are `mget` and `mput`, which allow multiple transfers, e.g.

```
mget data*
```

executes a `get` command for every file in the current directory whose name begins with `data`. Files cannot be renamed using `mget` or `mput`.

Usually you can interrupt a transfer using `ctrl/C`.

Other file operations

Various FTP programs provide some or all of the following:

`rm`	deletes a file on the remote system
`mkdir`	creates a directory on the remote system
`rmdir`	deletes a directory on the remote system
`chmod`	changes permissions on the remote system

To read a file (e.g. `/info/new`) on the remote system, use

```
get /info/new |more
```

If the remote system is a Unix system such as Pelican, Hermes or CUS on which you have an account, then you can issue a number of other useful commands by typing in a command preceded by `quote site`

For instance, to check your quota on the remote host type

```
quote site quota –v
```

To alter the default access permissions for your files on the remote system so that files you create in the current session are readable by everyone, type

```
quote site umask 022
```

To change access permissions on individual files, type a command such as

```
quote site chmod o+r *
```

To check access permissions, use the `dir` command.

Using FTP Explorer (Windows)

FTP Explorer is a Windows FTP client program available on the PWF. It is fairly self-explanatory and has its own help information, available from within the program by using the Help menu. An SCP client should be used in preference, except for anonymous FTP.

Start FTP Explorer from the Start menu (click on **Start**, point to **PWF Programs and Information**, **Communications and Networking**, then click on **FTP Explorer**). You should then see a dialog box entitled **Connect** (if this dialog box does not appear, select it from the **Tools** menu). The Host Address is shown as `cus.cam.ac.uk` by default; if you want to connect to anything other than CUS, fill in the Host Address box. Fill in your Login identifier if needed, and your password on the remote (host) system. For anonymous FTP you can just click on the **Anonymous** button, which automatically fills in `anonymous` as the user identifier and your e-mail address as the password. Then click on **Connect**.

If you are regularly going to connect to the same system, you can save the relevant information as a *profile* in your own filespace; fill in the details (including a Profile Name but not your password) on the Connect page and click on the **Save** button. The Profile Name will then appear in the list on the left of the Connect page, and you can select it again just by clicking on its name.

Once you are connected to your remote host, a list of folders and files will appear, with the full hierarchy on the left and the currently open folder on the right. You can navigate through the folders by (for instance) double-clicking on a folder name in the window to open it.

You can transfer files by dragging and dropping between the remote system and any open window on the desktop.

If you click the right mouse button on a remote file, you will be given the option of viewing, deleting, renaming or downloading the file; the last option is convenient if you want to download the file to a different name on your local system from that on the remote system. Viewing will only work if the system can find a suitable application for displaying the contents of the file (usually QuickView for text; binary files may not be viewable). The commands other than View are not available in an anonymous FTP session.

The program will normally decide whether the transfer should be ASCII or binary, based on the file type, but it may not always get this right; you can force the choice using the 

buttons on the Toolbar. You can designate certain file types which should always be transferred as text, or always transferred as binary, using the File Types section of the Options page (on the **View** menu).

If the remote system is a Unix system such as Pelican, Hermes or CUS on which you

have an account, then you can issue a number of other useful commands, by selecting **Quote** from the **Tools** menu and typing in a command preceded by `site`

For instance, to check your quota on the remote host type

```
site quota –v
```

To alter the default access permissions for your files on the remote system so that files you create in the current session are readable by everyone, type

```
site umask 022
```

To change access permissions on individual files, type a command such as

```
site chmod o+r *
```

which allows all other users to read any file in the current directory.

You can check access permissions in FTP Explorer; they should be on the list of files you normally see (you may need to scroll to the right within the window). If the files are displayed as icons or as a list without dates and permissions, then click on the Details button on the toolbar (to the left of the A/B buttons).

Using Fetch (Macintosh)

Fetch is the recommended FTP program on the Macintosh, and is available on the PWF. It is fairly self-explanatory and has its own help system, which can be opened independently of the program, or from within the program by using the **Windows** menu. MacSFTP, or the SCP facilities within NiftyTelnet, should be used in preference to Fetch except for anonymous FTP.

Start Fetch by double-clicking on the program icon as usual. You should then see a dialog box for filling in details of the system to connect to (if this dialog box does not appear, select **New Connection** from the **File** menu). Fill in the hostname (e.g. `cus`, `ftp.csx.cam.ac.uk`, etc.), the appropriate user identifier and password, and a directory name if you know which directory you need. For anonymous FTP, the Fetch program will fill in `anonymous` as the user identifier if you leave this field blank.

Then click on **OK**, and the connection should be made.

On your own Macintosh, you may want to store the connections you use most often as *Shortcuts*; select **New Shortcuts** from the **Customize** menu. Fill in the details (but note that you should not include your password in a shortcut specification) and click **OK**; you will then be able to open the connection at any time by selecting **Open Shortcut** from the **File** menu.

Once you are connected to your remote host, a list of files will appear in the left-hand box. You can look at other files, by double-clicking on a directory name or by using the **Directories** menu and selecting **Change Directory**.

To transfer a file, first select the mode of transfer. **Automatic** will work in most cases, but for a plain text file (not a formatted file such as a Word document), you may need to select **Text**, and for a formatted or compressed file you may need to select **Binary** (note that **Automatic** does not always select binary transfer when it should).

To fetch files, select the file or files to be transferred, and click on **Get File(s)**. If you are transferring one file only, you will be offered the usual dialog box to name the Macintosh file. If you are transferring several files, you will be asked which folder to put them in, but you do not have the option of changing the names.

Text files arriving at the Macintosh will by default (on the PWF) be created as Word files; they can however be read by any standard word processor including SimpleText.

To send files from the Macintosh to the remote host, use **Put File(s)**. The usual dialog box is provided to select the file to be sent, and a text box to specify the name on the host system. By default, for a text file, the name used is the Macintosh name with `.txt` added on the end. Various alternative formats (MacBinary II, BinHex, Text, Wrapped Text) may be selected for sending files; see Fetch Help for more details.

On most Macintosh systems you can also fetch and send files by dragging the file icon to the appropriate place.

The **Remote** menu in Fetch allows you to delete or rename files or directories on the remote system, and to view a remote file without explicitly fetching it. Most of these commands (other than View) are not available in an anonymous FTP session.

If the remote system is a Unix system such as Pelican, Hermes or CUS on which you have an account, then you can issue a number of other useful commands by selecting **Send FTP Command...** from the **Remote** menu and typing in a command preceded by site

For instance, to check your quota on the remote host type

```
site quota –v
```

To alter the default access permissions for your files on the remote system so that files you create in the current session are readable by everyone, type

```
site umask 022
```

To change access permissions on individual files, type a command such as

```
site chmod o+r *
```

which allows all other users to read any file in the current directory.

To check access permissions, use Fetch to show you the contents of the directory you're interested in, and then look at the full listing using the **`Fetch Transcript`** command from the **Windows** menu.

Anonymous FTP

This facility is used for fetching to your local system files which have been made publicly available on various *archive servers* all over the world. You do not need to have an account on the remote system, and you are not normally allowed to send files to it. It is usually more efficient to fetch files from local or UK or European sites wherever possible, rather than from the rest of the world. The Computing Service runs a server called `ftp.csx.cam.ac.uk` with a variety of local, public domain or University-licensed software. Beyond that, a primary Web site for finding out about, and fetching, software is `http://www.mirror.ac.uk/`

Finding out what files are available for fetching from a given server is easiest if the server provides a Web interface so that you can get the information using a Web browser. Otherwise you can use FTP to examine directories and so on. Web search engines can also be used to search the Internet for particular items of software

It is simplest, where possible, to fetch such files through the World Wide Web (using Netscape, Internet Explorer or other Web browsers); most of the major archive sites have Web interfaces with descriptive text, making it easy to select the files you need and download them with a mouse click.

Alternatively, in most Web browsers you can start an FTP session simply by entering a URL starting with ftp: instead of http:, for instance

```
ftp://ftp.csx.cam.ac.uk
```

for the Computing Service FTP server. This takes you to a directory listing from which you can navigate through directories to the file you want, and then double-click to download the file. Many browsers also allow you to fetch a file by dragging it from the browser window to any open folder on your desktop.

If neither of the above options is available to you, or if you need to do more complicated operations, then you may need to use a full FTP client, as described in below.

Whichever method you use, you may find that the files you fetch have been compressed or packed in some way; see below for more details about dealing with these.

Using an FTP client for anonymous FTP

Using an FTP client (e.g. ftp in Unix, FTP Explorer, or Fetch) for anonymous FTP is very like using it for anything else, except at the login stage. Full details are in earlier sections.

For anonymous FTP using an FTP program, many systems use the word `anonymous` as the user identifier; some use `guest` or `ftp` or `anon`. If your information about the archive server does not tell you what identifier to use, try these.

The convention for anonymous FTP is to type your e-mail address as a password (either the bare identifier `fjc1000` or the full address `fjc1000@cam.ac.uk`).

Note that with some combinations of FTP servers and clients (for instance a connection from CUS to the Computing Service FTP server) the `ls` command does not work as expected and the `dir` command should be used instead.

The other point to note about anonymous FTP is that most files (programs etc., and any file which has been compressed) will require to be transferred in binary (see above).

Example

To fetch from the Computing Service FTP server, using anonymous FTP, the README file about the Unix section of the server, and store it under a new name in subdirectory `info` on the client Unix system (e.g. PWF Linux)

`ftp ftp.csx.cam.ac.uk`	set up a connection to the FTP server
`Name: anonymous`	
`Password: fjc1000`	
`ftp> cd pub/unix`	point to the source directory on the server
`ftp> dir`	list the contents of the directory
`ftp> lcd info`	point to the destination directory on the PWF
`ftp> get README unixreadme`	fetch and rename the file
`ftp> quit`	close the connection and quit FTP

Compressed files

Often files which can be fetched using anonymous FTP have been compressed in some way. In Unix, compressed files can be recognised by the name which usually ends in `.gz` or `.z`. These files should be fetched in binary mode, and then uncompressed using the Unix `uncompress` or `gunzip` utility (see the man pages, or `camhelp compress` on CUS). The file may also be a tar file, in which case the tar utility is needed after the file has been uncompressed (see `camhelp tar` or `man tar`). More details can be found using `camhelp file transfer`.

PC files may also be compressed and need uncompressing before use. The type of compression is often (but not always) signalled by a special filename extension (for instance `.zip`). The principal unpacking utility on the PWF is Winzip (click on **Start**, move to **PWF Programs and Information**, then to **Utilities and Accessories**, and click on Winzip). It has its own Help system.

PC files may also be self-extracting archives (names ending in `.exe`) which need to be "run" to extract the individual files.

Macintosh files may be compressed; the Fetch program is usually configured to uncompress such files as they arrive, so no special separate stage is needed. Sometimes extra unpacking utilities (Stuffit Expander, for instance) may be needed, or the file may arrive as a "self-expanding archive" (name ending in `.sea`) which can be expanded simply by double-clicking on it.

Some details of special formats are also on the PWF Web pages at
`http://www.cam.ac.uk/localusersonly/cs/pwf/formats.html`

Appendix 1: FTP blocking

For security reasons the Computing Service blocks general access from outside the CUDN to FTP servers within. Specifically, connections from JANET into the CUDN to port 21 are blocked, except for a list of permitted CUDN hosts. This therefore affects users who need to provide an FTP server or similar facilities for use from outside Cambridge. The restrictions do not apply to SSH servers (for incoming SFTP or SCP calls).

FTP initiated by an FTP client running within Cambridge is not blocked, wherever the server may be. "Within Cambridge" includes machines (wherever they are) connected via the CUDN dial-up service (Magpie), so a user at home wishing to fetch files via Magpie from a Departmental machine is unaffected by the block.

The list of exceptions to the block includes the PWF, CUS, Hermes and Pelican.

The primary way round the block is to use an SSH server instead wherever possible, Users wishing to connect to the server then need to use an SFTP or SCP client, as described earlier in this leaflet.

Other alternatives are:

- use the CUDN dial-up service (Magpie) to avoid the problem
- arrange to initiate transfers using an FTP client within Cambridge (the block does not affect outgoing FTP).
- use an intermediate machine not subject to the block as a staging post.
- arrange that the relevant Cambridge system is adequately managed, both now and for the future, and apply through the appropriate institutional Computer Officer to cert@cam.ac.uk to have it included in the exceptions list.
- put the files on a Web server so that they can be accessed using HTTP.
- small files (e.g. plain text or Word documents of a few pages) can readily be sent by e-mail, or even on floppies, CDs, ZIP disks etc .

Further guidance is available at `http://www.cam.ac.uk/cs/support/secure/`

Appendix 2: Obtaining file transfer clients

Unix

Most Unix releases will include an FTP client and many will also provide for SFTP and/or SCP. Otherwise, see the details of the CD described below.

Windows

Windows releases will normally include an FTP client.

For secure transfer, see `http://www.chiark.greenend.org.uk/~sgtatham/putty/` for pscp, a line-mode add-on for PuTTY, and iXplorer, a graphical interface, and `http://winscp.vse.cz/eng/` for WinSCP. See also the free CD described below.

Macintosh

For information on obtaining Macintosh clients (Fetch, NiftyTelnet, MacSFTP), see `http://www-tus.csx.cam.ac.uk/mac_support/`

The SSH client CD

You can obtain a CD from Computing Service Reception (free of charge) with SSH and SCP clients for a range of platforms; this is strongly recommended if you are going to transfer files between systems outside the CUDN and systems within. The client programs on the CD can also be downloaded from `http://www-uxsup.csx.cam.ac.uk/CD`

Leaflet number: G72.

University of Cambridge Computing Service
New Museums Site
Pembroke Street
Cambridge
CB2 3QH

tel: **(01223) 334600**

fax: **(01223) 334679**

email: **reception@ucs.cam.ac.uk**