

File Transfer with the Versatile GFTP

FTP and Beyond

Whether you like your file transfers with a GUI or from a command prompt, GFTP has the right tool for every job. The GFTP client also supports advanced features such as secure file transfer with SSH. **BY MARTIN STEIGERWALD**

GFTP is a versatile and efficient File Transfer Protocol (FTP) client for Linux systems. GFTP, which was written by Brian Masney, performs all the standard chores associated with downloading and uploading files to FTP servers, and it also offers additional services, such as HTTP file transfer, SSH-based file transfer, and simultaneous download from multiple servers. The GFTP program provides several useful features that add efficiency to your FTP sessions, such as drag and drop, intelligent prompting for existing files, direct input for FTP commands, and direct transfer between two FTP servers.

GFTP comes in two flavors: the *gftp-gtk* variant supports convenient GUI-based file transfers, whereas *gftp-text* is a text-based FTP client used with the shell. If you type *gftp*, your system may launch either the X11 version or the text version, depending on how your system is configured.

GFTP is available for several popular Linux systems, although it is not always

installed automatically. See the box labeled “Installing GFTP” for more on installing GFTP with Suse or Debian, as well as details for building GFTP from source code.

Working with GFTP

GFTP launches a classical FTP-style split window, with panels showing the local and remote directories (see Figure 1). A status area below provides information on the current transfers, and another

area shows the current action for a protocol.

Enter the name of the FTP server to which you wish to connect in the box labeled *Host* at the top of the main window. You can also enter an FTP username and password, and you can specify the port number to use for the connection. The split window displays the local directory structure on the left and the remote directories on the right. When you find the file you wish to trans-

Installing GFTP

If you have Suse Linux, you can run *YaST* | *Install and remove software* to install the *gftp* package. SuSE Linux 9.1 has Version 2.0.16-44.

If you prefer to build the software yourself, you will need to install a few tools and developer files before you start. For example, in Suse, make sure you install the *YaST C/C++ Compiler and Tools* and *Gnome Development* selections.

Download the GFTP source code archive from the project homepage [1], and unpack the package by typing *tar -xjf gftp-2.0.17.tar.bz2*, or use desktop tools to unpack the tar archive. Change to the *gftp-2.0.17* directory, and follow the usual steps: *./configure ; make ; su -c ; make install*.

By default, *make install* places the executable GFTP files in */usr/local/bin*. This puts your custom-built version higher up the search path than the regular install, and this

also means that the menu items created by your package management system will point to your customized version. If you need to remove your own version, because Suse has introduced a new package, for example, you can enter *su -c “make uninstall”* to do so.

On Debian Linux, you can enter *su -c “apt-get install gftp”* or opt for a package manager such as Synaptic or KPackage to install GFTP; we tested GFTP with Debian *Sarge* version 2.0.17-5. The *gftp* package automatically installs both versions of GFTP, although you can install the individual packages, *gftp-gtk* and *gftp-text* separately if you prefer.

The packaged version for Debian does not provide SSL support at present (see [2]). If you intend to use FTPS or HTTPS, compile the software as described previously. But don't forget to install the *libssl-dev* package with the OpenSSL developer files before you do.

What about FTPS?

FTPS is an SSL-protected variant of FTP. GFTP supports the FTPS protocol for encrypting the control connection: the program transmits FTP commands with parameters such as the password as cipher text but does not encrypt files.



Figure 1: The GFTP main window provides a familiar FTP GUI for uploading and downloading files.

fer, select the file and click the appropriate arrow button. (The arrow buttons are located between the local and remote sides of the main window – see Figure 1. To transfer a file from the local to the remote directory, click the arrow pointing right. To transfer a file from the remote to the local directory, select the arrow pointing left.)

GFTP attempts to resolve file naming conflicts before the transfer begins. If a file with the same name already exists in the target directory, GFTP asks you what to do with the file (see Figure 2). The dialog allows you to specify an action for individual files or for all files. The program supports overwriting and has a continue mode for interrupted transfers. By default, GFTP suggests continuing a file transfer. You can select *Overwrite by default* in *FTP | Options | General* to change this behavior (see Figure 3).

The setting in the *Options | General* tab labeled *Allow command input in GUI* enables FTP command input (see Figure 3). This setting gives users the option of using a GUI without losing the benefit of direct command-line input. If this option is enabled, you can enter a text-style FTP

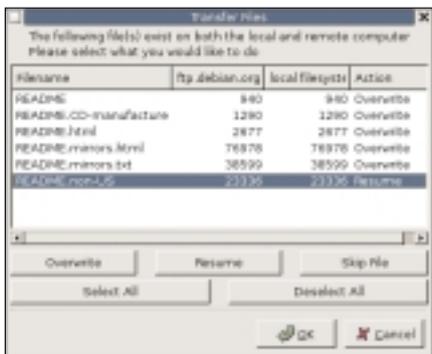


Figure 2: Resolving file name conflicts in GFTP.

command into the GFTP main window (see Figure 4).

If you wish to select multiple files from multiple directories to transfer all at once, you need to disable *Start transfer* to stop GFTP from running off and fetching the files. Then select the files you need, and re-enable the option to tell the program to start the file transfer.

GFTP's support for downloads from multiple services is excellent. Simply disconnect from the current server and connect to another server while the program carries on with the current download operation in the background.

If you select *Do transfer one at a time*, the FTP client will process the downloads one after the other. Otherwise, the program simultaneously loads one file each from each server where you have initiated a download. This only makes sense if you have a fast Internet connection.

GFTP also supports the direct transfer of files between two FTP servers. In other words, you can use FTP to move a

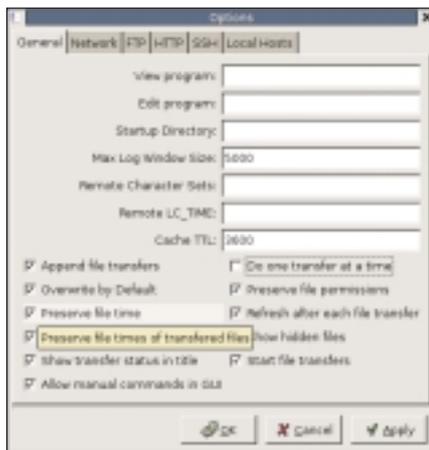


Figure 3: The General tab in the Options dialog box controls the program's general behavior.

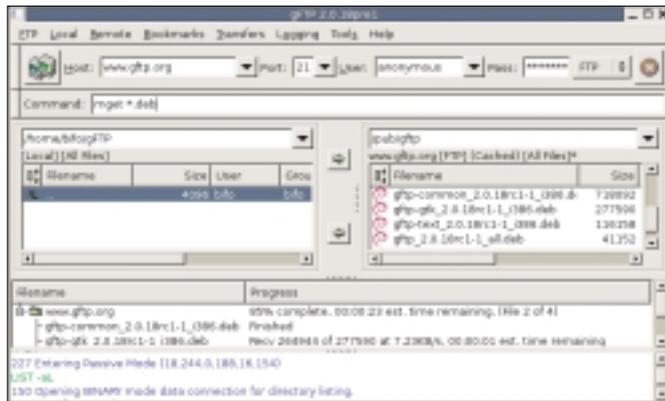


Figure 4: The best of both worlds: GFTP lets you use the mouse to navigate the target directory and then type a text-style FTP command.

file from one FTP server to another without ever storing the file on the GFTP client computer. GFTP's direct transfer option requires that both servers support the File Exchange Protocol (FXP), an extension of FTP. To launch a direct transfer session, connect to the FTP server that will be receiving the files. Then select *Local | Open URL...* to connect to the FTP server that will be sending the files.

If you attempt to terminate a connection while the program is establishing a connection or reading a directory, GFTP displays a message in the view area telling you to click on the *Stop* button first. If you do not notice the message, you might wonder why the program is not responding.

Configuring File Types

GFTP's configuration file allows you to specify which programs to use to display files of certain types. The program does not have a GUI-based configuration dialog for this. Open the file *.gftp/gftprc* in your home directory with your favorite editor (see Figure 5). Look for lines that start with *ext=*.

Each line follows the same pattern, starting with *ext=*, then a dot and the file extension, the XPM image file to use, the transmission mode, *B* for binary and *A* for ASCII, and finally the name, and if needed, the path to the program for displaying or playing the file type. Common filetypes like widely spread image and music file formats as well as HTML documents are predefined. Enter your favorite display program for these types. It is easy to create a new file type by copying and modifying an existing line.

GFTP in the Shell

If you need to transfer files automatically using a script, the shell variant of GFTP is a better option. Typing `help` gives you an overview of the commands supported by the GFTP shell version. You can type `help commandname` for more detail on a specific command.

You can type a command such as `gftp -d ftp://yourusername:yourpassword@your-domain.com/directory-with-html-files` to store the data for your web hosting account, and setting up a cronjob with this command would give you a regular backup.

Make sure you set restrictive privileges for your shell script to protect your FTP password. A script that contains a password should not be accessible to anyone but yourself. If in doubt, leave out the password. Of course, if you leave out the password, you would then need to type the password every time, which would eliminate the benefit of running the script with a cronjob.

Other Protocols

GFTP also supports file transfer through HTTP and SSH. For HTTP, type a URL that starts with `http://` in the *Host* box in the main window; for SSH, use a URL that starts with `ssh2://` instead. The program only supports version 2 of SSH.

There is no need to modify the default configurations for FTP and HTTP transmissions, unless you use a proxy server,

THE AUTHOR

Martin Steigerwald works as a system administrator at team(ix) GmbH in Nuremberg, Germany, <http://www.teamix.de>. One focus of his work is second level support for Linux as business desktop for customers of team(ix). He already installed Linux years ago on his Amiga 4000 and uses Linux – preferably Debian Linux – on his private computers.

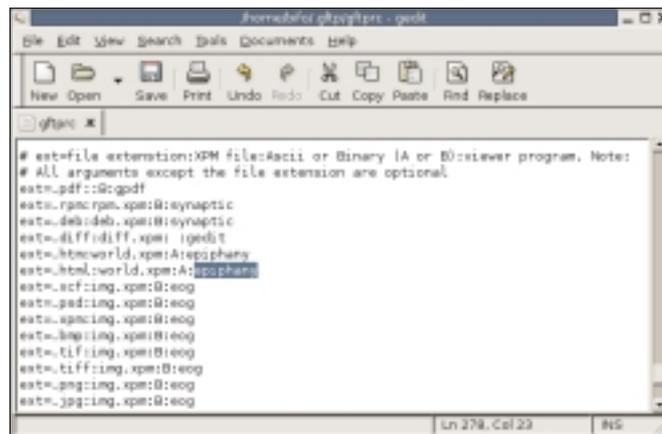


Figure 5: The configuration file allows you to map applications to file types.

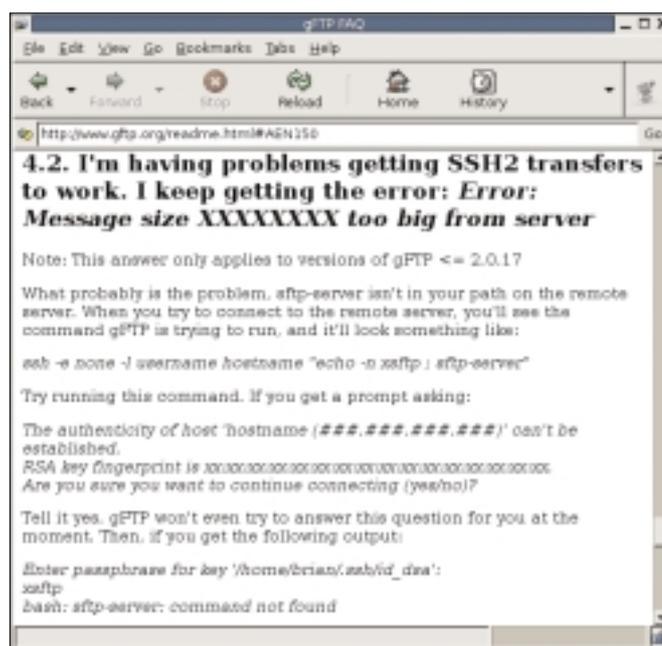


Figure 6: The GFTP FAQ answers questions on problems related to topics such as SSH connections.

but to make SSH transfers work, you need to change a few options below *FTP | Options | SSH*.

The FTP client uses SFTP as the transmission protocol. For this to work, the SFTP server needs to be in an executable path on the remote server. Enable the *Use SSH2 SFTP subsys* option for servers that support this.

We successfully tested both versions with an SSH server installation on Debian. The first connection attempt (using the SSH2 SFTP function) did not work because Secure Shell prompted us to confirm that we wanted to trust the server's authenticity. GFTP interprets the prompt as an error. But the remaining connections in this session were fine, as SSH only prompts you once a session.

If you intend to use SFTP for secure file transfers, you might like to wait until version 2.0.18 of GFTP is released. The current CVS snapshot for GFTP already resolves this problem, says the program's author. And there are bugfixes for other problems, whereas the current version of GFTP simply has workarounds in the FAQ [3]. For example, you no longer need to specify the path to the SFTP server. SFTP should work without any additional configuration. In a short test with pre-release version 2.0.18, an SSH connection worked with the default settings.

Secure SFTP-protected file transfers are definitely useful if you are uploading files to your own website. In contrast to an FTP client, an SFTP client encrypts the password and the data, however, this assumes that your provider has SSH access and SFTP support.

Conclusion

GFTP is an effective tool with a wide range of features. But there is still some room for improvement. The *Stop* buttons are not always intuitive. And the program would benefit

from GUI-based configuration of MIME type mappings and display programs. Support for the Free Desktop project's MIME database [4], would also be useful. ■

INFO

- [1] GFTP project website: <http://www.gftp.org>
- [2] GFTP binaries for Debian without SSL support: <http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=251121>
- [3] GFTP FAQ: <http://www.gftp.org/readme.html>
- [4] Shared MIME Database: <http://www.freedesktop.org/Standards/shared-mime-info-spec>
- [5] MIME actions: <http://www.freedesktop.org/Standards/mime-actions-spec>