



Transferring files with FTP

Sign up to the File Transfer Protocol and you'll be downloading patches, movies and distros with ease.



FILE TRANSFER: hardly the most accessible of topics, but it's something we all do on the web, and understanding it is really just a question of following protocol.

A protocol allows your computer to communicate with a remote computer by making sure both machines are using the same language. The most common protocol used these days is the Hypertext Transfer Protocol, or HTTP for short. You should recognise this as part of a typical website address, where it's simply telling your web browser to communicate with the other computer using HTTP. Files are made available as part of a website, and when you need to download one you can just click on a link and the download starts automatically.

Transferring files using HTTP is fine, but sometimes you need to transfer either a large number of files or ones that are particularly big. Many games provide updates called patches that might weigh in at several hundred megabytes. Then there are movies, audio files, and of course, Linux is often downloaded as a group of disc images than can be as large as several gigabytes.

For jobs like this, you'll need a different protocol.

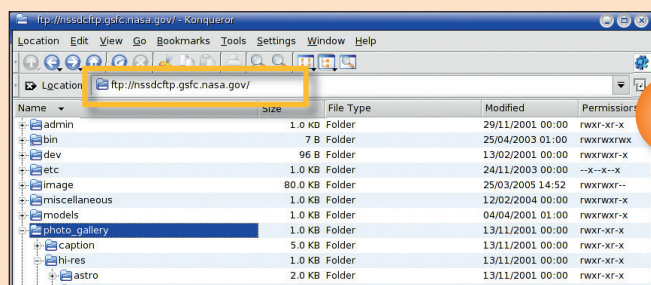
One of the oldest protocols for transferring files, predating much of the internet, is the concisely named File Transfer Protocol, or FTP. It's survived into the modern internet era because it's relatively easy to set up and use, and is perfect for downloading large files. FTP dates back to the old Unix networks, and was typically used from the shell as a command to access external sites. The site that holds the files is usually referred to as the server, whereas the users who connect and download the files are often referred to as the clients: a classic client-server model.

GRAPHICS MAKE IT EASY

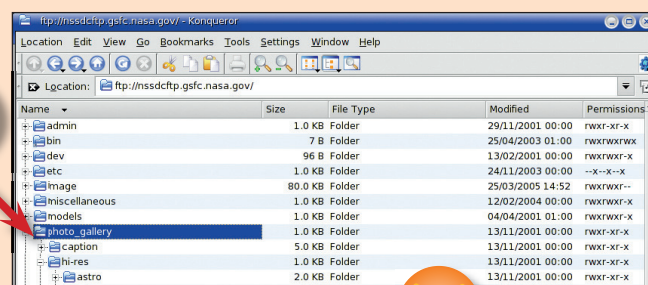
With Linux, you can still use the shell (also known as the command line) to download files using FTP, but it can be rather painful. More often than not, most people use a graphical client that hides the commands sent to the server, and some of the most popular clients are *gFTP* for Gnome and *KBear* for KDE. However, most web browsers support FTP directly, so there's often no need to use another



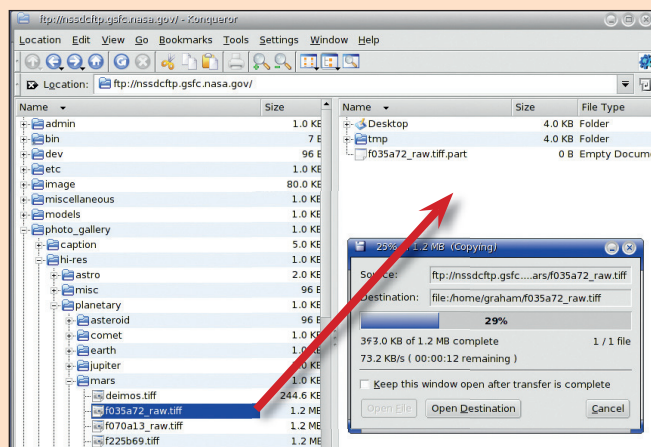
HOWTO... USE FTP TO DOWNLOAD FILES



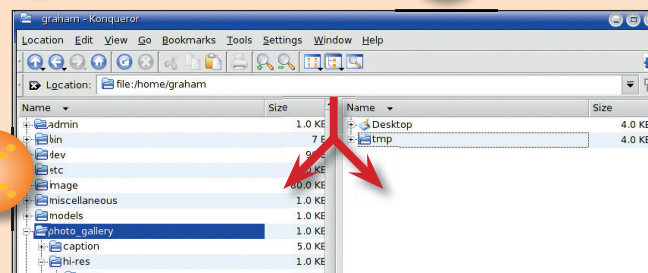
- 1 Open Konqueror and enter <ftp://nssdftp.gsfc.nasa.gov/> in the Location bar.



- 2 Navigate to **photo_gallery > hi-res > planetary > mars**.



- 4 Drag an image file from NASA's FTP site to your Home directory and it will be downloaded for you.



- 3 Split Konqueror's file display using **Window > Split View Left/Right**, then click on the Home icon.



program. This is particularly true of KDE's *Konqueror*, where you can take advantage of splitting the display or adding bookmarks.

When a URL contains the FTP protocol prefix (<ftp://>), *Konqueror* switches to a mode that looks almost identical to the one it uses for local file management. There's a good reason for this: *Konqueror* wants you to feel as comfortable navigating an external FTP site as you do browsing

command-line FTP program or 'client' such as *NcFTP*, which can automatically complete filenames, save a bookmark to get you back to the same location easily, and download using wild cards.

Wild cards, incidentally, are special characters (more often than not, simply `*`) used to imply a group of files. For example, `*.jpg` would indicate all files ending with `.jpg`. They are most often used in the shell, but the *FTP* command doesn't support

“Using FTP from the command line can be incredibly quick. You can easily use it to keep your system up to date.”

your own files. Directories and files appear exactly as local files do, and you can go backwards and forwards, up and down, in just the same way.

There are some differences from regular file management, though. One of them, and one of the best reasons for using FTP, depending on the server configuration, is that you can often resume an interrupted download. If, for some reason, you need to cancel a download, or you lose the connection to the server, just reconnect and copy the external file over the partly downloaded version. KDE will negotiate with the server to resume downloading the file from the point you reached before.

'ANONYMOUS' LOGIN

Most FTP sites you can access on the internet are anonymous. This doesn't mean that you can act with impunity; the site administrators will still be able to track your connection. The term 'anonymous' comes from the days when connections were made between Unix servers. To provide access to people who wanted to download files without having an account, a special kind of account was created where you would use 'anonymous' as your login name, and traditionally use your email address as the password. To keep the server secure, the account was usually read-only, so you couldn't delete files you weren't supposed to.

If you need to specify a password and username to access an FTP account, KDE will open a requester to ask for these details. But you can also enter them as part of the FTP site's URL, along with the directory you need to change to, as with <ftp://anonymous:password@ftpserver/directory>.

Using a URL like this has added advantages with KDE, because you can just bookmark the site using the Bookmark menu from *Konqueror*, and your user name will be saved automatically. *KWallet* also comes into play, as you can choose to save your user name and password in the wallet so that it's not open for everyone to see.

COMMAND-LINE TIPS

There are times when you have no alternative than to use the command line for FTP access, such as if you're restricted to a shell account, or you might want to use it just from preference. You can make the job a lot easier by using a modern

them. So you can't use `get *.jpg` with *FTP* for downloading all the JPEGs on the FTP site.

Here's an example of working with *NcFTP*: to open a connection with NASA's photo repository, you would type this from the command line:

```
$ ncfp ftp://nssdftp.gsfc.nasa.gov
```

If the connection is successful, you will see several pages of information from the remote site outlining its usage policy. Most command-driven FTP clients use the same commands for navigation, and *NcFTP* is no different. To change a remote directory, you use the `cd` command; to change your local working directory it's `lcd`. As in the shell, you can list file directories with `ls` for the remote site, or `lls` for the local one. So the next part of our code is:

```
ncftp / > cd /photo_gallery/hi-res/planetary/mars
ncftp .../hi-res/planetary/mars > ls
deimos.tiff hst_mars_9927e.tiff
f035a72_raw.tiff...
```

Downloading files is accomplished using the `get` command. The `ncftp` command is able to automatically complete filenames in the same way that `bash` can from the shell. For instance, to complete the `hst_mars_9927e.tiff` filename above, you would only need to type in 'hst_mars' and press Tab to complete the name. *NcFTP* also supports wild cards, so you could download every TIFF image from the NASA image repository by typing

```
ncftp .../hi-res/planetary/mars > get *.tiff
deimos.tiff: ETA: 0:02
100.00/244.60 kB 70.37 kB/s X
```

While it may seem a little arcane when compared with the many graphical clients, using FTP from the command line can be incredibly quick and useful, especially when used from within a script. You can easily use a few commands built around FTP to upload or download backup information from a remote site, or use it to keep your system up to date. To see a list of the other commands accessible from FTP, just type `help` while connected.

Using FTP, you can download whole groups of files from a remote site, or upload your own. It's often the only way of uploading a website, or downloading drivers, for example. Understanding a little more about how it works will help you to manage what can often be a large download, and can become a central part of how you use your internet connection. ●

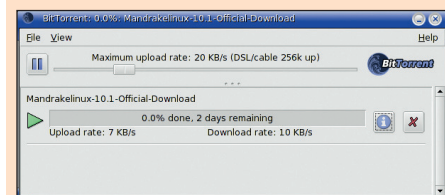
WILL I GET ARRESTED FOR USING BITTORRENT?

If you've heard of *BitTorrent*, you've almost certainly heard about it in connection with downloading illegal software from the internet. It's true that *BitTorrent* can be used this way, just as a telephone could be used by criminals, but the underlying technology actually has many legitimate uses.

The trouble with FTP is that it relies on a server, and a server costs money to run. There are hardware and maintenance costs, and the huge expense that comes from providing the bandwidth. At home, you probably have a cap on the maximum amount of data you can download with your ISP, typically several GB a month. That's the average size of a Linux distro, and dealing with hundreds or thousand of requests an hour can cripple a server, especially when the download is free.

The solution is to share the load, and that's what *BitTorrent* does. A download is split into lots of smaller chunks, typically smaller than a megabyte each, and every chunk you successfully download is shared with other people downloading the same file. In fact, you're downloading from people who are sharing their own data chunks, and as more and more people join, a torrent of data is streamed from a single starting point to the thousands of people all downloading and uploading separate chunks.

All this segmenting, uploading and downloading is transparent to you as a user. This means that the server, known as a tracker in torrent-speak, only has to provide the equivalent of a few copies before a torrent becomes self-sustaining.



BitTorrent in action, downloading an old version of a Linux distribution.

Before you download a file with *BitTorrent*, you first need to download a small 'torrent' file. This doesn't contain any data from the main file you want, but provides the information for the download agent. This includes the name and size of the file, along with a hash for each chunk that needs to be downloaded. A hash is basically a sequence of digits that can only be generated from the original file and is used to make sure that each chunk is a bit-for-bit replica of the source. The other important piece of information contained in a torrent file is the internet address of the tracker.

Once you've got hold of the torrent file, you can start *BitTorrent* from the shell by entering the following command:

```
$ btdownloadgui.py filetodownload.torrent
```

This opens a small window that contains the status of your download. It can take a few moments to get going; the software needs to connect to the tracker and download a list of peers who you can share with. Generally, the more you are able to share, the faster your download will proceed, which is why you can specify your connection speed with a slider to deal with times when you can spare the bandwidth, or times when you need to reduce it.

Download *BitTorrent* at www.bittorrent.com.