

# **GT 4.2.0 Component Guide to Public Interfaces: GridFTP**

DRAFT

## **GT 4.2.0 Component Guide to Public Interfaces: GridFTP**

DRAFT

---

# Table of Contents

1. API Summary .....	1
1. Programming Model Overview .....	1
2. Component API .....	1
I. GridFTP Commands .....	3
globus-url-copy .....	4
globus-gridftp-server .....	14
2. Graphical User Interface .....	24
3. Configuring GridFTP .....	25
1. GridFTP server configuration overview .....	25
2. Types of configurations .....	25
3. <b>globus-gridftp-server</b> quickstart .....	26
4. Running in daemon mode .....	27
5. Running under inetd or xinetd .....	27
4. Environment variable interface .....	29
1. Environment variables for GridFTP .....	29
A. Errors .....	30
Glossary .....	31

---

## List of Figures

1. Effect of Parallel Streams in GridFTP ..... 12

DRAFT

## List of Tables

1. URL formats .....	6
A.1. GridFTP Errors .....	30

DRAFT

# Chapter 1. API Summary

## 1. Programming Model Overview

The Globus FTP Client library provides a convenient way of accessing files on remote FTP servers. In addition to supporting the basic FTP protocol, the FTP Client library supports several security and performance extensions to make FTP more suitable for Grid applications. These extensions are described in the [GridFTP Protocol document](#)<sup>1</sup>.

In addition to protocol support for grid applications, the FTP Client library provides a [plugin architecture](#)<sup>2</sup> for installing application or grid-specific fault recovery and performance tuning algorithms within the library. Application writers may then target their code toward the FTP Client library and, by simply enabling the appropriate plugins, easily tune their application to run it on a different grid.

All applications which use the Globus FTP Client API must include the header file `globus_ftp_client.h` and activate the `GLOBUS_FTP_CLIENT_MODULE`<sup>3</sup>.

To use the Globus FTP Client API, one must create an [FTP Client handle](#)<sup>4</sup>. This structure contains:

- context information about FTP operations which are being executed,
- a cache of FTP control and data connections, and
- information about plugins which are being used.

The specifics of the connection caching and plugins are found in the "[Handle Attributes](#)"<sup>5</sup> section of the API documentation.

Once the handle is created, one may begin transferring files or doing other FTP operations by calling the functions in the "[FTP Operations](#)"<sup>6</sup> section of the API documentation. In addition to whole-file transfers, the API supports partial file transfers, restarting transfers from a known point, and various FTP directory management commands. All FTP operations may have a set of attributes, defined in the `operationattr` section, associated with them to tune various FTP parameters. The data structures and functions needed to restart a file transfer are described in the "[Restart Markers](#)"<sup>7</sup> section of the API documentation. For operations which require the user to send to or receive data from an FTP *server* they must call the functions described in the "`globus_ftp_client_data`" section of the manual.

The `globus_ftp_control` library provides low-level services needed to implement FTP clients and servers. The API provided is protocol specific. The data transfer portion of this API provides support for the standard data methods described in the FTP Specification as well as extensions for parallel, striped, and partial data transfer.

## 2. Component API

- [C Client Library API](#)<sup>8</sup>
- [C Control Library API](#)<sup>9</sup>

<sup>1</sup> <http://www.globus.org/alliance/publications/papers/GFD-R.0201.pdf>

<sup>2</sup> [http://www.globus.org/api/c-globus-4.2.0/globus\\_ftp\\_client/html/group\\_globus\\_ftp\\_client\\_plugins.html](http://www.globus.org/api/c-globus-4.2.0/globus_ftp_client/html/group_globus_ftp_client_plugins.html)

<sup>3</sup> [http://www.globus.org/api/c-globus-4.2.0/globus\\_ftp\\_client/html/group\\_globus\\_ftp\\_client\\_activation.html](http://www.globus.org/api/c-globus-4.2.0/globus_ftp_client/html/group_globus_ftp_client_activation.html)

<sup>4</sup> [http://www.globus.org/api/c-globus-4.2.0/globus\\_ftp\\_client/html/group\\_globus\\_ftp\\_client\\_handle.html](http://www.globus.org/api/c-globus-4.2.0/globus_ftp_client/html/group_globus_ftp_client_handle.html)

<sup>5</sup> [http://www.globus.org/api/c-globus-4.2.0/globus\\_ftp\\_client/html/group\\_globus\\_ftp\\_client\\_handleattr.html](http://www.globus.org/api/c-globus-4.2.0/globus_ftp_client/html/group_globus_ftp_client_handleattr.html)

<sup>6</sup> [http://www.globus.org/api/c-globus-4.2.0/globus\\_ftp\\_client/html/group\\_globus\\_ftp\\_client\\_operations.html](http://www.globus.org/api/c-globus-4.2.0/globus_ftp_client/html/group_globus_ftp_client_operations.html)

<sup>7</sup> [http://www.globus.org/api/c-globus-4.2.0/globus\\_ftp\\_client/html/group\\_globus\\_ftp\\_client\\_restart\\_marker.html](http://www.globus.org/api/c-globus-4.2.0/globus_ftp_client/html/group_globus_ftp_client_restart_marker.html)

<sup>8</sup> [http://www.globus.org/api/c-globus-3.9.x/globus\\_ftp\\_client/html/index.html](http://www.globus.org/api/c-globus-3.9.x/globus_ftp_client/html/index.html)

<sup>9</sup> [http://www.globus.org/api/c-globus-3.9.x/globus\\_ftp\\_control/html/index.html](http://www.globus.org/api/c-globus-3.9.x/globus_ftp_control/html/index.html)

For information on the internationalization API, see [Chapter 1, APIs](#).

DRAFT

---

# GridFTP Commands

DRAFT

# Name

globus-url-copy -- Multi-protocol data movement

globus-url-copy

## Tool description

**globus-url-copy** is a scriptable command line tool that can do multi-protocol data movement. It supports gsiftp:// (GridFTP), ftp://, http://, https://, and file:/// protocol specifiers in the URL. For GridFTP, globus-url-copy supports all implemented functionality. Versions from GT 3.2 and later support file globbing and directory moves.

- [Before you begin](#)
- [Command syntax](#)
- [Command line options](#)
  - [Informational options](#)
  - [Utility options](#)
  - [Reliability options](#)
  - [Performance options](#)
  - [Security-related options](#)
- [Default usage](#)
- [MODES in GridFTP](#)
- [If you run a GridFTP server by hand](#)
- [How do I choose a value for the TCP buffer size \(-tcp-bs\) option?](#)
- [How do I choose a value for the parallelism \(-p\) option?](#)
- [Limitations](#)
- [Interactive clients for GridFTP](#)

## Before you begin

### Important

To use gsiftp:// and https:// protocols, you must have a [certificate](#) to use globus-url-copy. However, you may use ftp:// or http:// protocols without a certificate.

1. First, as with all things Grid, you *must* have a valid proxy certificate to run globus-url-copy in certain protocols (gsiftp:// and https://, as noted above). If you are using ftp:// or http:// protocols, security is *not* mandatory and you may skip the rest of this table.

If you do not have a certificate, you must [obtain one](#).

If you are doing this for testing in your own environment, the [SimpleCA](#) provided with the Globus Toolkit should suffice.

If not, you must contact the Virtual Organization (VO) with which you are associated to find out whom to ask for a certificate.

One common source is the [DOE Science Grid CA](#)<sup>1</sup>, although you must confirm whether or not the resources you wish to access will accept their certificates.

Instructions for proper installation of the certificate should be provided from the source of the certificate.

Please note when your certificates expire; they will need to be renewed or you may lose access to your resources.

- Now that you have a certificate, you must generate a temporary proxy. Do this by running:

```
grid-proxy-init
```

Further documentation for **grid-proxy-init** can be found [here](#).

- You are now ready to use **globus-url-copy**! See the following sections for syntax and command line options and other considerations.

## Command syntax

The basic syntax for **globus-url-copy** is:

```
globus-url-copy [optional command line switches] Source_URL Destination_URL
```

where:

[optional command line switches]	See <a href="#">Command line options</a> below for a list of available options.
<i>Source_URL</i>	Specifies the original URL of the file(s) to be copied. If this is a directory, all files within that directory will be copied.
<i>Destination_URL</i>	Specifies the URL where you want to copy the files. If you want to copy multiple files, this must be a directory.

### Note

Any url specifying a directory must end with `/`.

## URL prefixes

As of GT 3.2, we support the following URL prefixes:

- file://** (on a local machine only)
- ftp://**
- gsiftp://**
- http://**

<sup>1</sup> <http://www.doe grids.org/pages/cert-request.htm>

- **https://**

By default, **globus-url-copy** expects the same kind of host certificates that **globusrun** expects from gatekeepers.



## Note

We do *not* provide an interactive client similar to the generic FTP client provided with Linux. See the [Interactive Clients](#) section below for information on an interactive client developed by NCSA/NMI/TeraGrid.

## URL formats

URLs can be any valid URL as defined by RFC 1738 that have a [protocol](#) we support. In general, they have the following format: ***protocol://host:port/path***.



## Note

If the path ends with a trailing / (i.e. `/path/to/directory/`) it will be considered to be a directory and all files in that directory will be moved. If you want a recursive directory move, you need to add the `-r/-recurse` switch described below.

**Table 1. URL formats**

<code>gsiftp://myhost.mydomain.com:2812/data/foo.dat</code>	Fully specified.
<code>http://myhost.mydomain.com/mywebpage/default.html</code>	Port is not specified; therefore, GridFTP uses protocol default (in this case, 80).
<code>file:///foo.dat</code>	Host is not specified; therefore, GridFTP uses your local host. Port is not specified; therefore, GridFTP uses protocol default (in this case, 80).
<code>file:/foo.dat</code>	This is also valid but is not recommended because, while many servers (including ours) accept this format, it is <i>not</i> RFC conformant and is not recommended.



## Important

For GridFTP (`gsiftp://`) and FTP (`ftp://`), it is legal to specify a user name and password in the the URL as follows:

```
gsiftp://myname:[mypassword]@myhost.mydomain.com/foo.dat
```

If you are using GSI security, then you may specify the username (but you may *not* include the `:` or the password) and the grid-mapfile will be searched to see if that is a valid account mapping for your distinguished name (DN). If it is found, the *server* will setuid to that account. If not, it will fail. It will NOT fail back to your default account.

If you are using anonymous FTP, the username *must* be one of the usernames listed as a valid anonymous name and the password can be anything.

If you are using password authentication, you must specify both your username and password. **THIS IS HIGHLY DISCOURAGED, AS YOU ARE SENDING YOUR PASSWORD IN THE CLEAR ON THE NETWORK.** This is worse than no security; it is a false illusion of security.

## Command line options

### Informational Options

-help   -usage	Prints help.
-version	Prints the version of this program.
-versions	Prints the versions of all modules that this program uses.
-q   -quiet	Suppresses all output for successful operation.
-vb   -verbose	During the transfer, displays: <ul style="list-style-type: none"><li>• number of bytes transferred,</li><li>• performance since the last update (currently every 5 seconds), and</li><li>• average performance for the whole transfer.</li></ul>
-dbg   -debugftp	Debugs FTP connections and prints the entire control channel protocol exchange to STDERR.  Very useful for debugging. Please provide this any time you are requesting assistance with a globus-url-copy problem.
-list <url>	This option will display a directory listing for the given url.

### Utility Ease of Use Options

-a   -ascii	Converts the file to/from ASCII format to/from local file format.
-b   -binary	Does not apply any conversion to the files. This option is turned on by default.
-f <i>filename</i>	Reads a list of URL pairs from a filename.  Each line should contain:  <i>sourceURL destURL</i>  Enclose URLs with spaces in double quotes ("). Blank lines and lines beginning with the hash sign (#) will be ignored.
-r   -recurse	Copies files in subdirectories.
-notpt   -no-third-party-transfers	Turns third-party transfers off (on by default).  Site firewall and/or software configuration may prevent a connection between the two servers (a <i>third party transfer</i> ). If this is the case, globus-url-copy will "relay" the data. It will do a GET from the source and a PUT to the destination.  This obviously causes a performance penalty but will allow you to complete a transfer you otherwise could not do.

### Reliability Options

-rst   -restart	Restarts failed FTP operations.
-----------------	---------------------------------

-rst-retries <retries>	Specifies the maximum number of times to retry the operation before giving up on the transfer.  Use 0 for infinite.  The default value is 5.
-rst-interval <seconds>	Specifies the interval in seconds to wait after a failure before retrying the transfer.  Use 0 for an exponential backoff.  The default value is 0.
-rst-timeout <seconds>	Specifies the maximum time after a failure to keep retrying.  Use 0 for no timeout.  The default value is 0.

## Performance Options

-tcp-bs <size>   -tcp-buffer-size <size>	Specifies the size (in bytes) of the TCP buffer to be used by the underlying ftp data channels.
--	---

### Important

This is critical to good performance over the WAN.

#### How do I pick a value?

-p <parallelism>   -parallel <parallelism>	Specifies the number of parallel data connections that should be used.
--	--

### Note

This is one of the most commonly used options.

#### How do I pick a value?

-bs <block size>   -block-size <block size>	Specifies the size (in bytes) of the buffer to be used by the underlying transfer methods.
---	--

-pp	<b>(New starting with GT 4.1.3)</b> Allows pipelining. GridFTP is a command response protocol. A client sends one command and then waits for a "Finished response" before sending another. Adding this overhead on a per-file basis for a large data set partitioned into many small files makes the performance suffer. Pipelining allows the client to have many outstanding, unacknowledged transfer commands at once. Instead of being forced to wait for the "Finished response" message, the client is free to send transfer commands at any time.
-----	--

-mc <i>filename source_url</i>	<b>(New starting with GT 4.2.0)</b> Transfers a single file to many destinations. File-name is a line-separated list of destination urls. For more information on this option, click <a href="#">here</a> .
--------------------------------	---

Multicasting must be enabled for use on the server side.

## Security Related Options

`-s <subject> | -subject <subject>` Specifies a subject to match with both the source and destination servers.



### Note

Used when the server does not have access to the host certificate (usually when you are running the server as a user). See [the section called “If you run a GridFTP server by hand...”](#).

`-ss <subject> | -source-subject <subject>` Specifies a subject to match with the source server.



### Note

Used when the server does not have access to the host certificate (usually when you are running the server as a user). See [the section called “If you run a GridFTP server by hand...”](#).

`-ds <subject> | -dest-subject <subject>` Specifies a subject to match with the destination server.



### Note

Used when the server does not have access to the host certificate (usually when you are running the server as a user). See [the section called “If you run a GridFTP server by hand...”](#).

`-nodcau | -no-data-channel-authentication` Turns off data channel authentication for FTP transfers (the default is to authenticate the data channel).



### Warning

We do *not* recommend this option, as it is a security risk.

`-dcsafe | -data-channel-safe` Sets data channel protection mode to SAFE.

Otherwise known as *integrity* or *checksumming*.

Guarantees that the data channel has not been altered, though a malicious party may have observed the data.



### Warning

Rarely used as there is a substantial performance penalty.

`-dcpriv | -data-channel-private` Sets data channel protection mode to PRIVATE.

The data channel is encrypted and checksummed.

Guarantees that the data channel has not been altered and, if observed, it won't be understandable.

 **Warning**

VERY rarely used due to the VERY substantial performance penalty.

## Default globus-url-copy usage

A **globus-url-copy** invocation using the **gsift** protocol with no options (i.e., using all the defaults) will perform a transfer with the following characteristics:

- binary
- stream mode (which implies no parallelism)
- host default TCP buffer size
- encrypted and checksummed control channel
- an authenticated data channel

## MODES in GridFTP

GridFTP (as well as normal FTP) defines multiple wire protocols, or MODES, for the data channel.

Most normal FTP servers only implement *stream mode* (MODE S), i.e. the bytes flow in order over a single TCP connection. GridFTP defaults to this mode so that it is compatible with normal FTP servers.

However, GridFTP has another MODE, called Extended Block Mode, or *MODE E*. This mode sends the data over the data channel in blocks. Each block consists of 8 bits of flags, a 64 bit integer indicating the offset from the start of the transfer, and a 64 bit integer indicating the length of the block in bytes, followed by a payload of length bytes. Because the offset and length are provided, out of order arrival is acceptable, i.e. the 10th block could arrive before the 9th because you know explicitly where it belongs. This allows us to use multiple TCP channels. If you use the `-p 1` | `-parallelism` option, **globus-url-copy** automatically puts the servers into MODE E.



### Note

Putting `-p 1` is not the same as no `-p` at all. Both will use a single stream, but the default will use stream mode and `-p 1` will use MODE E.

## If you run a GridFTP server by hand...

If you run a GridFTP server by hand, you will need to explicitly specify the subject name to expect. The subject option provides **globus-url-copy** with a way to validate the remote servers with which it is communicating. Not only must the server trust **globus-url-copy**, but **globus-url-copy** must trust that it is talking to the correct server. The validation is done by comparing host DNs or subjects.

If the GridFTP server in question is running under a host certificate then the client assumes a subject name based on the server's canonical DNS name. However, if it was started under a user certificate, as is the case when a server is started by hand, then the expected subject name must be explicitly stated. This is done with the `-ss`, `-sd`, and `-s` options.

`-ss` Sets the `sourceURL` subject.

`-ds` Sets the `destURL` subject.

- s If you use this option alone, it will set both urls to be the same. You can see an example of this usage under the [Troubleshooting](#) section.



## Note

This is an *unusual* use of the client. Most times you need to specify both URLs.

## How do I choose a value?

### How do I choose a value for the TCP buffer size (-tcp-bs) option?

The value you should pick for the TCP buffer size (-tcp-bs) depends on how fast you want to go (your bandwidth) and how far you are moving the data (as measured by the Round Trip Time (RTT) or the time it takes a packet to get to the destination and back).

To calculate the value for -tcp-bs, use the following formula (this assumes that Mega means 1000<sup>2</sup> rather than 1024<sup>2</sup>, which is typical for bandwidth):

$$-tcp-bs = \text{bandwidth in Megabits per second (Mbs)} * \text{RTT in milliseconds (ms)} * 1000 / 8$$

As an example, if you are using fast ethernet (100 Mbs) and the RTT was 50 ms it would be:

$$-tcp-bs = 100 * 50 * 1000 / 8 = 625,000 \text{ bytes.}$$

So, how do you come up with values for bandwidth and RTT? To determine RTT, use either ping or traceroute. They both list RTT values.



## Note

You must be on one end of the transfer and ping the other end. This means that if you are doing a third party transfer you have to run the ping or traceroute between the two server hosts, not from your client.

The bandwidth is a little trickier. Any point in the network can be the bottleneck, so you either need to talk with your network engineers to find out what the bottleneck link is or just assume that your host is the bottleneck and use the speed of your network interface card (NIC).



## Note

The value you pick for -tcp-bs limits the top speed you can achieve. You will NOT get bandwidth any higher than what you used in the calculation (assuming the RTT is actually what you specified; it varies a little with network conditions). So, if for some reason you want to limit the bandwidth you get, you can do that by judicious choice of -tcp-bs values.

So where does this formula come from? Because it uses the bandwidth and the RTT (also known as the latency or delay) it is called the *bandwidth delay product*. The very simple explanation is this: TCP is a reliable protocol. It must save a copy of everything it sends out over the network until the other end acknowledges that it has been received.

As a simple example, if I can put one byte per second onto the network, and it takes 10 seconds for that byte to get there, and 10 seconds for the acknowledgment to get back (RTT = 20 seconds), then I would need at least 20 bytes of storage. Then, hopefully, by the time I am ready to send byte 21, I have received an acknowledgement for byte 1 and I can free that space in my buffer. If you want a more detailed explanation, try the following links on TCP tuning:

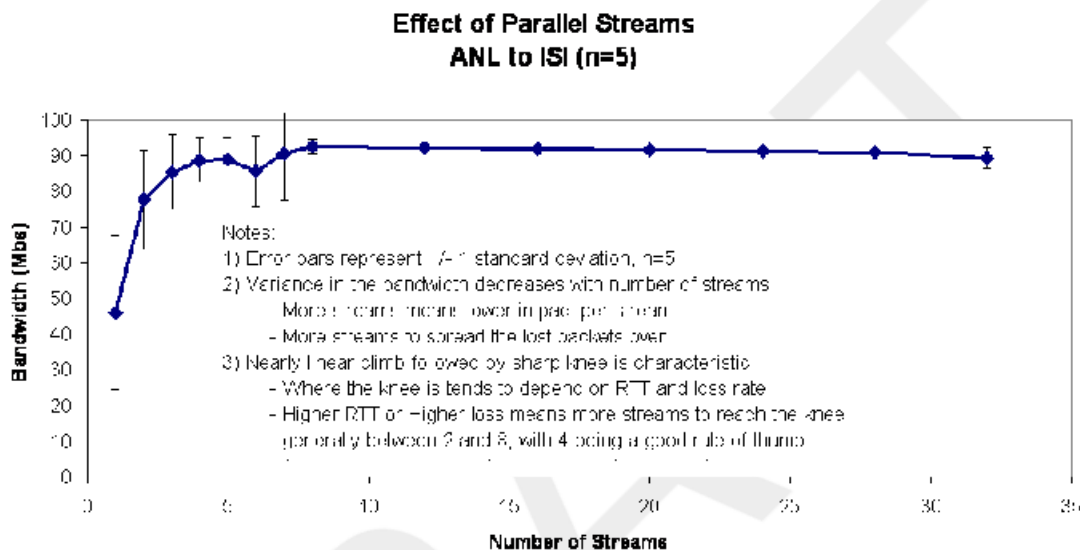
- [http://www.psc.edu/networking/perf\\_tune.html](http://www.psc.edu/networking/perf_tune.html)

- <http://www.didc.lbl.gov/TCP-tuning/>
- <http://www.ncne.nlanr.net/research/tcp/>

## How do I choose a value for the parallelism (-p) option?

For most instances, using 4 streams is a very good rule of thumb. Unfortunately, there is not a good formula for picking an exact answer. The shape of the graph shown here is very characteristic.

**Figure 1. Effect of Parallel Streams in GridFTP**



You get a strong, nearly linear, increase in bandwidth, then a sharp knee, after which additional streams have very little impact. Where this knee is depends on many things, but it is generally between 2 and 10 streams. Higher bandwidth, longer round trip times, and more congestion in the network (which you usually can only guess at based on how applications are behaving) will move the knee higher (more streams needed).

In practice, between 4 and 8 streams are usually sufficient. If things look really bad, try 16 and see how much difference that makes over 8. However, anything above 16, other than for academic interest, is basically wasting resources.

## Limitations

There are no limitations for **globus-url-copy** in GT 4.2.0.

## Interactive clients for GridFTP

The Globus Project does *not* provide an interactive client for GridFTP. Any normal FTP client will work with a GridFTP server, but it cannot take advantage of the advanced features of GridFTP. The interactive clients listed below take advantage of the advanced features of GridFTP.

There is no endorsement implied by their presence here. We make no assertion as to the quality or appropriateness of these tools, we simply provide this for your convenience. We will *not* answer questions, accept bugs, or in any way shape or form be responsible for these tools, although they should have mechanisms of their own for such things.

UberFTP was developed at the NCSA under the auspices of NMI and TeraGrid:

- NCSA Uerftp only download: <http://dims.ncsa.uiuc.edu/set/uberftp/download.html>
- UberFTP User's Guide: <http://dims.ncsa.uiuc.edu/set/uberftp/userdoc.html>

DRAFT

# Name

globus-gridftp-server -- Configures the GridFTP Server

globus-gridftp-server

## Tool description

**globus-gridftp-server** configures the GridFTP server using a config file and/or commandline options.



### Note

Command line options and configuration file options may both be used, but the command line *overrides* the config file.

The configuration file for the GridFTP *server* is read from the following locations, in the given order. Only the first file found will be loaded:

- Path specified with the `-c <configfile>` command line option.
- `$GLOBUS_LOCATION/etc/gridftp.conf`
- `/etc/grid-security/gridftp.conf`

Options are one per line, with the format:

```
<option> <value>
```

If the value contains spaces, they should be enclosed in double-quotes ("). Flags or boolean options should only have a value of 0 or 1. Blank lines and lines beginning with # are ignored.

For example:

```
port 5000
allow_anonymous 1
anonymous_user bob
banner "Welcome!"
```

## Developer notes

The Globus implementation of the GridFTP *server* draws on:

- three IETF RFCs:
  - RFC 959
  - RFC 2228
  - RFC 2389
- an IETF Draft: MLST-16
- the GridFTP protocol specification, which is Global Grid Forum (GGF) Standard GFD.020.

The command line tools and the *client* library completely hide the details of the protocol from the user and the developer. Unless you choose to use the control library, it is not necessary to have a detailed knowledge of the protocol.

## Command syntax

The basic syntax for **globus-gridftp-server** is:

```
globus-gridftp-server [optional command line switches]
```

To use **globus-gridftp-server** with a config file, make sure to use the `-c <configfile>` option.

## Command line options

The table below lists config file options, associated command line options (if available) and descriptions.

### Note

Any boolean option can be negated on the command line by preceding the specified option with '-no-' or '-n'.  
example: -no-cas or -nf.

## Informational Options

help <0 1>, -h, -help	Show usage information and exit. Default value: FALSE
version <0 1> , -v, -version	Show version information for the server and exit. Default value: FALSE
versions <0 1>, -v, -versions	Show version information for <b>all</b> loaded globus libraries and exit. Default value: FALSE

## Modes of Operation

inetd <0 1>, -i, -inetd	Run under an inetd service. Default value: FALSE
daemon <0 1>, -s, -daemon	Run as a daemon. All connections will fork off a new process and setuid if allowed. See <a href="#">Section 4, “Running in daemon mode”</a> for more information. Default value: TRUE
detach <0 1>, -S, -detach	Run as a background daemon detached from any controlling terminals. See <a href="#">Section 4, “Running in daemon mode”</a> for more information. Default value: FALSE
exec <string> , -exec <string>	For statically compiled or non-GLOBUS_LOCATION standard binary locations, specify the full path of the server binary here. Only needed when run in <a href="#">daemon mode</a> . Default value: not set

<code>chdir &lt;0 1&gt;, -chdir</code>	Change directory when the server starts. This will change directory to the dir specified by the <code>chdir_to</code> option.  Default value: TRUE
<code>chdir_to &lt;string&gt;, -chdir-to &lt;string&gt;</code>	Directory to <code>chdir</code> to after starting. Will use <code>/</code> if not set.  Default value: not set
<code>fork &lt;0 1&gt;, -f, -fork</code>	Server will fork for each new connection. Disabling this option is only recommended when debugging. Note that non-forked servers running as 'root' will only accept a single connection and then exit.  Default value: TRUE
<code>single &lt;0 1&gt;, -1, -single</code>	Exit after a single connection.  Default value: FALSE

## Authentication, Authorization, and Security Options

<code>auth_level &lt;number&gt;, -auth-level &lt;number&gt;</code>	<ul style="list-style-type: none"><li>• 0 = Disables all authorization checks.</li><li>• 1 = Authorize identity only.</li><li>• 2 = Authorize all file/resource accesses.</li></ul> <p>If not set, the GridFTP Server uses level 2 for front ends and level 1 for data nodes.</p> <p>Default value: not set</p>
<code>allow_from &lt;string&gt;, -allow-from &lt;string&gt;</code>	Only allow connections from these source IP addresses. Specify a comma-separated list of IP address fragments. A match is any IP address that starts with the specified fragment. Example: '192.168.1.' will match and allow a connection from 192.168.1.45. Note that if this option is used, any address not specifically allowed will be denied.  Default value: not set
<code>deny_from &lt;string&gt;, -deny-from &lt;string&gt;</code>	Deny connections from these source IP addresses. Specify a comma-separated list of IP address fragments. A match is any IP address that starts with the specified fragment. Example: '192.168.2.' will match and deny a connection from 192.168.2.45.  Default value: not set
<code>cas &lt;0 1&gt;, -cas</code>	Enable <u>Community Authorization Service (CAS)</u> authorization. For complete instructions on setting up a GridFTP server to use CAS, click <a href="#">here</a> .  Default value: TRUE
<code>secure_ipc &lt;0 1&gt;, -si, -secure-ipc</code>	Use GSI security on the IPC channel.  Default value: TRUE

<code>secure_ipc</code> <code>&lt;0 1&gt;</code> , <code>-si</code> , <code>-secure-ipc</code>	Use GSI security on the IPC channel. Default value: TRUE
<code>ipc_auth_mode</code> <code>&lt;string&gt;</code> , <code>-ia</code> <code>&lt;string&gt;</code> , <code>-ipc-auth-</code> <code>mode &lt;string&gt;</code>	Set GSI authorization mode for the IPC connection. Options are one of the following: <ul style="list-style-type: none"><li>• none</li><li>• host</li><li>• self</li><li>• subject:[subject]</li></ul> Default value: host
<code>allow_anonym-</code> <code>ous &lt;0 1&gt;</code> , <code>-aa</code> , <code>-allow-</code> <code>anonymous</code>	Allow cleartext anonymous access. If server is running as root, <code>anonymous_user</code> must also be set. Disables IPC security. Default value: FALSE
<code>anonym-</code> <code>ous_names_al-</code> <code>lowed</code> <code>&lt;string&gt;</code> , <code>-an-</code> <code>onymous-</code> <code>names-allowed</code> <code>&lt;string&gt;</code>	Comma-separated list of names to treat as anonymous users when allowing anonymous access. If not set, the default names of 'anonymous' and 'ftp' will be allowed. Use '*' to allow any user-name. Default value: not set
<code>anonym-</code> <code>ous_user</code> <code>&lt;string&gt;</code> , <code>-an-</code> <code>onymous-user</code> <code>&lt;string&gt;</code>	User to setuid to for an anonymous connection. Only applies when running as root. Default value: not set
<code>anonym-</code> <code>ous_group</code> <code>&lt;string&gt;</code> , <code>-an-</code> <code>onymous-group</code> <code>&lt;string&gt;</code>	Group to setgid to for an anonymous connection. If not set, the default group of <code>anonymous_user</code> will be used. Default value: not set
<code>pw_file</code> <code>&lt;string&gt;</code> , <code>-password-</code> <code>file &lt;string&gt;</code>	Enable cleartext access and authenticate users against this <code>/etc/passwd</code> formatted file. Default value: not set
<code>connec-</code> <code>tions_max</code> <code>&lt;number&gt;</code> , <code>-connections-</code> <code>max &lt;number&gt;</code>	Maximum concurrent connections allowed. Only applies when running in <u>daemon mode</u> . Unlimited if not set. Default value: not set
<code>connec-</code> <code>tions_dis-</code> <code>abled &lt;0 1&gt;</code> ,	Disable all new connections. Does not affect ongoing connections. This must be set in the configuration file and then a SIGHUP issued to the server in order to reload the configuration. Default value: FALSE

-connections-  
disabled

## Logging Options

`log_level` Log level. A comma-separated list of levels from the following:

`<string>`, `-d`  
`<string>`,  
`-log-level`  
`<string>`

- ERROR
- WARN
- INFO
- DUMP
- ALL

For example:

```
globus-gridftp-server -d error,warn,info
```

You may also specify a numeric level of 1-255.

Default value: ERROR

`log_module` Indicates the `globus_logging` module that will be loaded. If not set, the default `stdio` module will be used and the logfile options (see next option) will apply.

`<string>`,  
`-log-module`  
`<string>`

Built-in modules are `stdio` and `syslog`. Log module options may be set by specifying `module:opt1=val1:opt2=val2`. Available options for the built-in modules are:

- `interval` - Indicates buffer flush interval. Default is 5 seconds. A 0 second flush interval will disable periodic flushing, and the buffer will only flush when it is full.
- `buffer` - Indicates buffer size. Default is 64k. A value of 0k will disable buffering and all messages will be written immediately.

Example:

```
-log-module stdio:buffer=4096:interval=10
```

Default value: not set

`log_single` Indicates the path of a single file to which you want to log all activity. If neither this option nor `log_unique` is set, logs will be written to `stderr`, unless the execution mode is detached, or `inetd`, in which case logging will be disabled.

`<string>`, `-l`  
`<string>`,  
`-logfile`  
`<string>`

Default value: not set

`log_unique` Partial path to which `gridftp.(pid).log` will be appended to construct the log filename. Example:

`<string>`, `-L`  
`<string>`, `-logdir` `<string>`

```
-L /var/log/gridftp/
```

will create a separate log (`/var/log/gridftp/gridftp.xxxx.log`) for each process (which is normally each new *client* session). If neither this option nor `log_single` is set, logs will be written to `stderr`, unless the execution mode is detached, or `inetd`, in which case logging will be disabled.

	Default value: not set
log_transfer <string>, -Z <string>, -log-transfer <string>	Log NetLogger-style info for each transfer into this file. Default value: not set Example: DATE=20050520163008.306532 HOST=localhost PROG=globus-gridftp-server NL.EVNT=FTP_INFO START=20050520163008.305913 USER=ftp FILE=/etc/group BUF- FER=0 BLOCK=262144 NBYTES=542 VOLUME=/ STREAMS=1 STRIPES=1 DEST=[127.0.0.1] TYPE=RETR CODE=226 Time format is YYYYMMDDHHMMSS.UUUUUU (microsecs). <ul style="list-style-type: none"> <li>• DATE: time the transfer completed.</li> <li>• START: time the transfer started.</li> <li>• HOST: hostname of the server.</li> <li>• USER: username on the host that transferred the file.</li> <li>• BUFFER: tcp buffer size (if 0 system defaults were used).</li> <li>• BLOCK: the size of the data block read from the disk and posted to the network.</li> <li>• NBYTES: the total number of bytes transferred.</li> <li>• VOLUME: the disk partition where the transfer file is stored.</li> <li>• STREAMS: the number of parallel TCP streams used in the transfer.</li> <li>• STRIPES: the number of stripes used on this end of the transfer.</li> <li>• DEST: the destination host.</li> <li>• TYPE: the transfer type, RETR is a send and STOR is a receive (ftp 959 commands).</li> <li>• CODE: the FTP rfc959 completion code of the transfer. 226 indicates success, 5xx or 4xx are failure codes.</li> </ul>
log_filemode <string>, -log-filemode <string>	File access permissions of log files. Should be an octal number such as 0644 (the leading 0 is required). Default value: not set
disable_us- age_stats <0 1>, -dis- able-usage- stats	Disable transmission of per-transfer usage statistics. See the <a href="#">Usage Statistics</a> <sup>1</sup> section in the online documentation for more information. Default value: FALSE
us- age_stats_tar- get <string>, -usage-stats-	Comma-separated list of contact strings for usage statistics listeners. The format of <string> is host:port. Default value: usage-stats.globus.org:4810

<sup>1</sup> ../../Usage\_Stats.html

**target**            **Example:**  
 <string>  
                   -usage-stats-target usage-stats.globus.org:4810,usage-stats.uc.teragrid.org

In this example, the usage statistics will be transmitted to the default Globus target (usage-stats.globus.org:4810) and another target (usage-stats.uc.teragrid.org:5920).

## Single and Striped Remote Data Node Options

**remote\_nodes**    Comma-separated list of remote node contact strings. See [Remote data-nodes and striped operations](#) and [Separation of processes for higher security](#) for examples of using this option.

<string>, -r  
 <string>, -remote-nodes  
 <string>  
 Default value: not set

**data\_node**        This server is a back end data node. See [Separation of processes for higher security](#) for an example of using this option.

<0|1>, -dn,  
 -data-node  
 Default value: FALSE

**stripe\_blocksize** Size in bytes of sequential data that each stripe will transfer.

<number>,  
 -sbs <number>  
 , -stripe-blocksize  
 <number>  
 Default value: 1048576

**stripe\_layout**    Stripe layout. 1 = Partitioned, 2 = Blocked.

<number>, -sl  
 <number>,  
 -stripe-layout  
 <number>  
 Default value: 2

**stripe\_blocksize\_locked** Do not allow client to override stripe blocksize with the **OPTS RETR** command.

<0|1>,  
 -stripe-blocksize-locked;  
 Default value: FALSE

**stripe\_layout\_locked** Do not allow client to override stripe layout with the **OPTS RETR** command.

<0|1>,  
 -stripe-layout-locked  
 Default value: FALSE

## Disk Options

**blocksize**        Size in bytes of data blocks to read from disk before posting to the network.

<number>, -bs  
 <number>,  
 -blocksize  
 <number>  
 Default value: 262144

`sync_writes` `<0|1>`, `-sync-writes` Flush disk writes before sending a restart marker. This attempts to ensure that the range specified in the restart marker has actually been committed to disk. This option will probably impact performance and may result in different behavior on different storage systems. See the man page for `sync()` for more information.

Default value: FALSE

## Network Options

`port` `<number>`, `-p` `<number>`, `-port` `<number>` Port on which a front end will listen for client control channel connections or on which a data node will listen for connections from a front end. If not set, a random port will be chosen and printed via the logging mechanism. See [Remote data-nodes and striped operations](#) and [Separation of processes for higher security](#) for examples of using this option.

Default value: not set

`control_interface` `<string>`, `-control-interface` `<string>` Hostname or IP address of the interface to listen for control connections on. If not set, will listen on all interfaces.

Default value: not set

`data_interface` `<string>`, `-data-interface` `<string>` Hostname or IP address of the interface to use for data connections. If not set will use the current control interface.

Default value: not set

`ipc_interface` `<string>`, `-ipc-interface` `<string>` Hostname or IP address of the interface to use for IPC connections. If not set, will listen on all interfaces.

Default value: not set

`hostname` `<string>`, `-hostname` `<string>` Effectively sets the above `control_interface`, `data_interface` and `ipc_interface` options.

Default value: not set

`ipc_port` `<number>`, `-ipc-port` `<number>` Port on which the front end will listen for data node connections.

Default value: not set

## Timeouts

`control_preauth_timeout` `<number>`, `-control-preauth-timeout` `<number>` Time in seconds to allow a client to remain connected to the control channel without activity before authenticating.

Default value: 30

`control_idle_timeout` `<number>`; `-control-` Time in seconds to allow a client to remain connected to the control channel without activity.

Default value: 600

idle-timeout  
<number>

ipc\_idle\_timeout Idle time in seconds before an unused IPC connection will close.  
<number> ,  
-ipc-idle- Default value: 600  
timeout <num-  
ber>

ipc\_con- Time in seconds before cancelling an attempted IPC connection.  
nect\_timeout  
<number> , Default value: 60  
-ipc-connect-  
timeout <num-  
ber>

## User Messages

banner Message that is displayed to the client before authentication.  
<string> ,  
-banner Default value: not set  
<string>

banner\_file Read banner message from this file.  
<string> ,  
-banner-file Default value: not set  
<string>

banner\_terse When this is set, the minimum allowed banner message will be displayed to unauthenticated clients.  
<0|1> , -ban-  
ner-terse  
Default value: FALSE

login\_msg Message that is displayed to the client after authentication.  
<string> , -lo-  
gin-msg  
<string>  
Default value: not set

lo- Read login message from this file.  
gin\_msg\_file  
<string> , -lo-  
gin-msg-file  
<string>  
Default value: not set

## Module Options

load\_dsi\_mod- Load this Data Storage Interface module. File and remote modules are defined by the server. If  
ule <string> , not set, the file module is loaded, unless the remote option is specified, in which case the remote  
-dsi <string> module is loaded. An additional configuration string can be passed to the DSI using the format  
[module name]:[configuration string]. The format of the configuration string is  
defined by the DSI being loaded.

Default value: not set

`allowed_modules <string>` Comma-separated list of ERET/ESTO modules to allow and, optionally, specify an alias for.  
Example:  
`, -allowed-modules <string>` `-allowed-modules module1,alias2:module2,module3`  
(module2 will be loaded when a client asks for alias2).  
Default value: not set

## Other Options

`configfile <string>, -c <string>` Path to configuration file that should be loaded. Otherwise will attempt to load `$GLOBUS_LOCATION/etc/gridftp.conf` and `/etc/grid-security/gridftp.conf`.  
Default value: not set

`use_home_dirs <0|1>, -use-home-dirs` Set the startup directory to the authenticated user's home dir.  
Default value: TRUE

`debug <0|1>, -debug` Set options that make the server easier to debug. Forces no-fork, no-chdir, and allows core dumps on bad signals instead of exiting cleanly. Not recommended for production servers. Note that non-forked servers running as root will only accept a single connection and then exit.  
Default value: FALSE

## Limitations

For transfers using parallel data transport streams and for transfers using multiple computers at each end, the direction of the connection on the data channels must go from the sending to the receiving side. For more information about this limitations see <http://www.ogf.org/documents/GFD.20.pdf>.

Globus GridFTP server does not run on windows

---

## Chapter 2. Graphical User Interface

Globus does not provide any interactive client for GridFTP, either GUI or text based. However, NCSA, as part of their TeraGrid activity, produces a text based interactive client called UberFTP, which you may want to check out. See [the section called “Interactive clients for GridFTP”](#) for more information.

DRAFT

# Chapter 3. Configuring GridFTP

## 1. GridFTP server configuration overview

The configuration interface for GridFTP is the admin tool, [globus-gridftp-server](#), which can be used with a configuration file and/or run-time options.

### Note

Command line options and configuration file options may both be used, but the command line *overrides* the config file.

The configuration file for the GridFTP *server* is read from the following locations, in the given order. Only the first file found will be loaded:

- Path specified with the `-c <configfile>` command line option.
- `$GLOBUS_LOCATION/etc/gridftp.conf`
- `/etc/grid-security/gridftp.conf`

Options are one per line, with the format:

`<option> <value>`

If the value contains spaces, they should be enclosed in double-quotes ("). Flags or boolean options should only have a value of 0 or 1. Blank lines and lines beginning with # are ignored.

For example:

```
port 5000
allow_anonymous 1
anonymous_user bob
banner "Welcome!"
```

For complete command documentation including all options, see [globus-gridftp-server\(1\)](#).

This page includes information about general configuration of the GridFTP server. Security options are discussed [here](#), and more advanced configuration is described [here](#).

## 2. Types of configurations

The following describes different GridFTP configurations of the front end (control channel) and back end (data channels).

1. **Typical configuration:** this is the default where the data channel and control channel are separate socket connections within the same process. The client sends a command and waits to finish before issuing the next command. This is good for a single host, traditional-type user. If you have a single host and you want an ultra-reliable and light weight file transfer service, this is a good choice. Also good for testing purposes.
2. **Separate processes (or split process):** control channel and data channel are on different ports - with front end run as a non-privileged user (typically the `globus` user) with very limited access to the machine and the back end is run as root, but configured to only allow connections from the front end from a local machine. This means

an external user is never connected to a root running process and thus minimizes the impact of an exploit. This does, however, require that a copy of the [host cert and host key](#) be owned by the non-privileged user. If you use this configuration, the non-privileged user should not have write permission to executables, configuration files, etc. Provides greater security and also allows for proxying and load balancing. Many backend data movers can be behind a single point of client contact. Each client is assigned a different backend in a round robin fashion. For more information about this configuration, see [Section 2, “Separation of Processes \(Split Process\)”](#).

3. **Striped servers:** single control channel (front end), multiple data channels (back end) This is recommended for improved performance of large (1GB+) file transfers. Can also be useful if you want to use full data encryption and need to tether together many hosts to handle the processing load. For more information about this configuration, see [Section 1, “Remote data-nodes and striped operation”](#).



## Note

Furthermore, #2 and #3 can be combined. You can have an 8 node cluster that only uses 2 nodes at a time in a striped server configuration and load balances across the rest of the nodes.

# 3. globus-gridftp-server quickstart

The following is a quick guide to running the server and using the client:

Look through the list of options for globus-gridftp-server:

```
globus-gridftp-server --help
```

Start the server in anonymous mode (discussed more fully [here](#)):

```
globus-gridftp-server -control-interface 127.0.0.1 -aa -p 5000
```

where:

`-control-interface` is the hostname or IP address of the interface to listen for control connections on . This option is only needed here as a rudimentary means of security for this simple example.

`-aa` enables anonymous mode

`-p` indicates on which port the server listens.

Run a two party transfer with client:

```
globus-url-copy -v file:///etc/group ftp://localhost:5000/tmp/group
```

Run 3rd party transfer:

```
globus-url-copy -v ftp://localhost:port/etc/group ftp://localhost:port/tmp/group2
```

Experiment with `-dbg`, and `-vb` options for debugging and checking the performance of your setup:

```
globus-url-copy -dbg file:///etc/group ftp://localhost:5000/tmp/group
```

```
globus-url-copy -vb file:///dev/zero ftp://localhost:5000/dev/null
```

where:

`-dbg` A useful option when something is not working. It results in a GridFTP control channel protocol dump (along with other useful information) to stderr. If you understand the GridFTP protocol, or you have ambition to

understand it, this can be a very useful tool to discover various problems in your setup such as overloaded servers and firewalls. When submitting a bug report or asking a question on the support email lists one should always send along the `-dbg` output.

`-vb` Provides a type of progress bar of the user to observe the rate at which their transfer is progressing.

Ctrl-c - Kill the server.

### Note

There are many possible options and configurations with **globus-gridftp-server**. For some guidelines on setting it up for your situation, see [Chapter 5, Key Admin Settings and Tuning Recommendations](#).

## 4. Running in daemon mode

The server should generally be run as root in daemon mode, although it is possible to run it as a user (see below). When run as root you will need to have a [host certificate](#).

Run the server:

```
globus-gridftp-server < -s | -S > <args>
```

where:

- `-s` Runs in the foreground (this is the default mode).
- `-S` Detaches from the terminal and runs in the background.

The following additional steps may be required when running as a user other than root (for more details, review [Basic Security Configuration](#)):

- Create a `~/ .gridmap` file, containing the DNs of any clients you wish to allow, mapped to the current username.
- Create a proxy with **grid-proxy-init**.

## 5. Running under inetd or xinetd

### Note

We also feature a user-configurable, super-server daemon plugin called GFork. Click [here](#) for more information.

### 5.1. Set up xinetd/inetd config file

#### Note

The service name used (`gsiftp` in this case) should be defined in `/etc/services` with the desired port.

Here is a sample GridFTP server xinetd config entry in `/etc/xinetd.conf`:

```
service gsiftp
{
instances          = 100
socket_type        = stream
```

```
wait                = no
user                = root
env                 += GLOBUS_LOCATION=(globus_location)
env                 += LD_LIBRARY_PATH=(globus_location)/lib
server              = (globus_location)/sbin/globus-gridftp-server
server_args         = -i
log_on_success      += DURATION
nice                = 10
disable             = no
}
```

Here is a sample gridftp server inetd config entry in `/etc/inetd.conf` (read as a single line):

```
gsiftp stream tcp nowait root /usr/bin/env env \
  GLOBUS_LOCATION=(globus_location) \
  LD_LIBRARY_PATH=(globus_location)/lib \
  (globus_location)/sbin/globus-gridftp-server -i
```



### Note

On Mac OS X, you must set `DYLD_LIBRARY_PATH` instead of `LD_LIBRARY_PATH` in the above examples.

On IRIX, you may need to set either `LD_LIBRARYN32_PATH` or `LD_LIBRARY64_PATH`.

## 5.2. globus-gridftp-server -i

Use the `-i` commandline option with `globus-gridftp-server`:

```
globus-gridftp-server -i
```

# Chapter 4. Environment variable interface

## 1. Environment variables for GridFTP

The GridFTP *server* or *client* libraries do not read any environment variable directly, but the security and networking related variables described below may be useful.

- [Non-WS \(General\) Authentication & Authorization Environment Variables.](#)
- [XIO Network Driver Environment Variables.](#)

DRAFT

# Appendix A. Errors

**Table A.1. GridFTP Errors**

Error Code	Definition	Possible Solutions
<pre>globus_ftp_client: the server responded with an error 530 530-glo- bus_xio: Authentication Error 530-OpenSSL Error: s3_srvr.c:2525: in lib- rary: SSL routines, function SSL3_GET_CLI- ENT_CERTIFICATE: no cer- tificate returned 530- globus_gsi_callback_mod- ule: Could not verify credential 530-glo- bus_gsi_callback_module: Can't get the local trusted CA certificate: Untrusted self-signed certificate in chain with hash d1b603c3 530 End.</pre>	<p>This error message indicates that the GridFTP server doesn't trust the certificate authority (CA) that issued your certificate.</p>	<p>You need to ask the GridFTP server administrator to install your CA certificate chain in the GridFTP server's trusted certificates directory.</p>
<pre>globus_ftp_control: gss_init_sec_context failed OpenSSL Error: s3_clnt.c:951: in lib- rary: SSL routines, function SSL3_GET_SERV- ER_CERTIFICATE: certific- ate verify failed glo- bus_gsi_callback_module: Could not verify creden- tial globus_gsi_call- back_module: Can't get the local trusted CA certificate: Untrusted self-signed certificate in chain with hash d1b603c3</pre>	<p>This error message indicates that your local system doesn't trust the certificate authority (CA) that issued the certificate on the resource you are connecting to.</p>	<p>You need to ask the resource administrator which CA issued their certificate and install the CA certificate in the local trusted certificates directory.</p>

# Glossary

## C

**client** A process that sends commands and receives responses. Note that in GridFTP, the client may or may not take part in the actual movement of data.

## E

**extended block mode (MODE E)** MODE E is a critical GridFTP components because it allows for out of order reception of data. This in turn, means we can send the data down multiple paths and do not need to worry if one of the paths is slower than the others and the data arrives out of order. This enables parallelism and striping within GridFTP. In MODE E, a series of “blocks” are sent over the data channel. Each block consists of:

- an 8 bit flag field,
- a 64 bit field indicating the offset in the transfer,
- and a 64 bit field indicating the length of the payload,
- followed by length bytes of payload.

Note that since the offset and length are included in the block, out of order reception is possible, as long as the receiving side can handle it, either via something like a seek on a file, or via some application level buffering and ordering logic that will wait for the out of order blocks.

## S

**server** A process that receives commands and sends responses to those commands. Since it is a server or service, and it receives commands, it must be listening on a port somewhere to receive the commands. Both FTP and GridFTP have IANA registered ports. For FTP it is port 21, for GridFTP it is port 2811. This is normally handled via inetd or xinetd on Unix variants. However, it is also possible to implement a daemon that listens on the specified port. This is described more fully in the Architecture section of the GridFTP Developer's Guide.

**stream mode (MODE S)** The only mode normally implemented for FTP is MODE S. This is simply sending each byte, one after another over the socket in order, with no application level framing of any kind. This is the default and is what a standard FTP server will use. This is also the default for GridFTP.

## T

**third party transfers** In the simplest terms, a third party transfer moves a file between two GridFTP servers.

The following is a more detailed, programmatic description.

In a third party transfer, there are three entities involved. The client, who will only orchestrate, but not actually take place in the data transfer, and two servers one of which will be sending data to the other. This scenario is common in Grid applications where you may wish to stage data from a data store somewhere to a super-computer you have reserved. The commands are quite similar to the client/server transfer. However, now the client must establish two control channels, one to each server. He will then choose one to listen, and send it the PASV command. When it responds with the IP/port it is listening on, the client will send that IP/port as part of the PORT command to the other server. This will cause the second server to connect to the first server, rather than the client. To initiate the actual movement of the data, the client then sends the RETR “filename” command to the server that will read from disk and write to the network (the “sending” server) and will send the STOR “filename” command to the other server which will read from the network and write to the disk (the “receiving” server).  
See Also [client/server transfer](#).