

GT 5.0.2 Component Guide to Public Interfaces: GridFTP

GT 5.0.2 Component Guide to Public Interfaces: GridFTP

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-url-sync	16
globus-gridftp-server	19
2. Graphical User Interface	31
1. Globus GridFTP GUI (pre-alpha)	31
2. UberFTP	31
3. Configuring GridFTP	32
1. GridFTP server configuration overview	32
2. Typical configuration	32
3. Firewall requirements	33
4. Configuring Security for GridFTP	34
5. globus-gridftp-server quickstart	38
4. Environment variable interface	40
1. Environment variables for GridFTP	40
A. Errors	41
Glossary	44

List of Figures

1. Effect of Parallel Streams in GridFTP	15
--	----

List of Tables

1. URL formats	6
2. URL formats	17
A.1. GridFTP Errors	42

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](#)¹.

In addition to protocol support for grid applications, the FTP Client library provides a [plugin architecture](#)² 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`³.

To use the Globus FTP Client API, one must create an [FTP Client handle](#)⁴. 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](#)"⁵ 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](#)"⁶ 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](#)"⁷ 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](#)⁸
- [C Control Library API](#)⁹

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

² http://www.globus.org/api/c-globus-5.0.2/globus_ftp_client/html/group_globus_ftp_client_plugins.html

³ http://www.globus.org/api/c-globus-5.0.2/globus_ftp_client/html/group_globus_ftp_client_activation.html

⁴ http://www.globus.org/api/c-globus-5.0.2/globus_ftp_client/html/group_globus_ftp_client_handle.html

⁵ http://www.globus.org/api/c-globus-5.0.2/globus_ftp_client/html/group_globus_ftp_client_handleattr.html

⁶ http://www.globus.org/api/c-globus-5.0.2/globus_ftp_client/html/group_globus_ftp_client_operations.html

⁷ http://www.globus.org/api/c-globus-5.0.2/globus_ftp_client/html/group_globus_ftp_client_restart_marker.html

⁸ http://www.globus.org/api/c-globus-3.9.x/globus_ftp_client/html/index.html

⁹ 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](#).

GridFTP Commands

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 in `globus-url-copy`, you must have a [certificate](#). However, you may use `ftp://`, `http://` or `sshftp://` 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://`, `http://` or `sshftp://` protocols, you may skip ahead to [Command syntax](#)

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](#)¹, 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 Command line options 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

Versions from GT 3.2 and later support the following URL prefixes:

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

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

- **https://**

Versions from GT 4.2 and later support the following URL prefix (in addition to the above-mentioned URL prefixes):

- **sshftp://**



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"> • number of bytes transferred, • performance since the last update (currently every 5 seconds), and • average performance for the whole transfer.
-dbg -debugftp	<p>Debugs FTP connections and prints the entire control channel protocol exchange to STDERR.</p> <p>Very useful for debugging. Please provide this any time you are requesting assistance with a globus-url-copy problem.</p>
-list <url>	This option will display a directory listing for the given url.
-nl-bottleneck -nlb	This option uses NetLogger to estimate speeds of disk and network read/write system calls, and attempt to determine the bottleneck component.



Note

In order to use this, the server must be configured to [enable netlogger bottleneck detection](#)².

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.
-cd -create-dest	Create destination directories, if needed
-f <i>filename</i>	<p>Reads a list of URL pairs from a filename.</p> <p>Each line should contain:</p> <p><i>sourceURL destURL</i></p>

² <http://www.cedps.net/index.php/Gridftp-netlogger>

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 *third party transfer*). 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.

-df <filename> | -dumpfile <filename>

Specifies path to the file where untransferred urls will be saved for later restarting. The resulting file is the same format as the -f input file. If the file exists, it will be read and all other url input will be ignored.

-stall-timeout | -st <seconds>

Specifies how long before cancelling/restarting a transfer with no data movement. Set to 0 to disable. Default is 600 seconds.

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 **(New starting with GT 4.1.3)** 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 *filename source_url* > Transfers a single file to many destinations. Filename is a line-separated list of destination urls. For more information on this option, click [here](#).

Multicasting must be [enabled for use](#) on the server side.

 **Warning**

This option is EXPERIMENTAL.

-concurrency | -cc Specifies the number of concurrent FTP connections to use for multiple transfers.

-udt Uses UDT, a reliable UDP-based transport protocol, for data transfers.

-fast Recommended when using GridFTP servers. Use MODE E for all data transfers, including reusing data channels between list and transfer operations.

Note: In order to use this option, the server must be configured to use [UDT](#). For third party transfers, no change is required on the client side. For client-server transfers, you need the threaded flavor of the client. Refer to [Switching between threaded and non-threaded flavors](#) for information on how to switch between threaded and non-threaded flavors of globus-url-copy.

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.

Advanced Options


-stripe



Enables striped transfers on supported servers.

-striped-block-size | -sbs

Sets layout mode and blocksize for striped transfers.

If not set, the server defaults will be used.

	If set to 0, partitioned mode will be used.
	If set to >0, blocked mode will be used, with this setting used as the blocksize.
-t <transfer time in seconds>	Runs the transfer for the specified number of seconds and then ends. Useful for performance testing or forced restart loops.
-ipv6	Uses ipv6 when available.
 Warning	
This option is EXPERIMENTAL. Use at your own risk.	
-dp -delayed-pasv	Enables delayed passive.
-g2 -gridftp2	Uses GridFTP v2 protocol enhancements when possible.
-mn -module-name <gridftp storage module name>	Specifies the backend storage module to use for both the source and destination in a GridFTP transfer.
-mp -module-parameters <gridftp storage module parameters>	Specifies the backend storage module arguments to use for both the source and destination in a GridFTP transfer.
-smn -src-module-name <gridftp storage module name>	Specifies the backend storage module to use for the source file in a GridFTP transfer.
-smp -src-module-parameters <gridftp storage module parameters>	Specifies the backend storage module arguments to use for the source file in a GridFTP transfer.
-dmn -dst-module-name <gridftp storage module name>	Specifies the backend storage module to use for the destination file in a GridFTP transfer.
-dmp -dst-module-parameters <gridftp storage module parameters>	Specifies the backend storage module arguments to use for the destination file in a GridFTP transfer.
-aa -authz-assert <authorization assertion file>	Uses the assertions in the specified file to authorize access to both the source and destination servers.
-saa -src-authz-assert <authorization assertion file>	Uses the assertions in the specified file to authorize access to the source server.
-daa -dst-authz-assert <authorization assertion file>	Uses the assertions in the specified file to authorize access to the destination server.
-cache-aa -cache-authz-assert	Caches the authorization assertion for subsequent transfers.
-cache-saa -cache-src-authz-assert	Caches the source authorization assertion for subsequent transfers.
-cache-daa -cache-dst-authz-assert	Caches the destination authorization assertion for subsequent transfers.
-nl-bottleneck -nlb	Uses NetLogger to estimate speeds of disk and network read/write system calls, and attempt to determine the bottleneck component.
	Note: In order to use this, the server must be configured to enable netlogger bottleneck detection.

-src-pipe -SP <command line>	<p>Sets the source end of a remote transfer to use piped-in input with the given command line.</p> <p> Warning</p> <p>Do not use with the <code>-fsstack</code> option.</p>
-dst-pipe -DP <command line>	<p>Sets the destination end of a remote transfer to write data to then standard input of the program run via the given command line.</p> <p> Warning</p> <p>Do not use with the <code>-fsstack</code> option.</p>
-pipe <command line>	Sets both <code>-src-pipe</code> and <code>-dst-pipe</code> to the same value.
-dcstack -data-channel-stack	Specifies the XIO driver stack for the network on both the source and the destination. Both must be GridFTP servers.
-fsstack -file-system-stack	Specifies the XIO driver stack for the disk on both the source and the destination. Both must be GridFTP servers.
-src-dcstack -source-data-channel-stack	Specifies the XIO driver stack for the network on the source GridFTP server.
-src-fsstack -source-file-system-stack	Specifies the XIO driver stack for the disk on the source GridFTP server.
-dst-dcstack -dest-data-channel-stack	Specifies the XIO driver stack for the network on the destination GridFTP server.
-dst-fsstack -dest-file-system-stack	Specifies the XIO driver stack for the disk on the destination GridFTP server.
-cred <path to credentials or proxy file>, -src-cred -sc <path to credentials or proxy file>, -dst-cred -dc <path to credentials or proxy file>	Specifies the credentials to use for source, destination, or both FTP connections.
-af <filename> -alias-file <filename>	Specifies a file that maps logical host aliases to lists of physical hosts. When used with multiple concurrent connections, each connection uses the next host in the list. Each line should either be an alias (noted with the @ symbol), or a hostname[:port]. Currently, only the aliases @source and @destination are valid, and they are used for every source or destination url.

Synchronization Options

-sync	Only transfer files where the destination does not exist or differs from the source. -sync-level controls how to determine if files differ.
-sync-level <number>	Choose criteria for determining if files differ when performing a sync transfer. Level 0 will only transfer if the destination does not exist. Level 1 will transfer if the size of the destination does not match the size of the source. Level 2 will transfer if the timestamp of the destination is older than the timestamp of the source. Level 3 will

perform a checksum of the source and destination and transfer if the checksums do not match. The default sync level is 2.

Default globus-url-copy usage

A **globus-url-copy** invocation using the **gsiftp** 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 1000^2 rather than 1024^2 , 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.

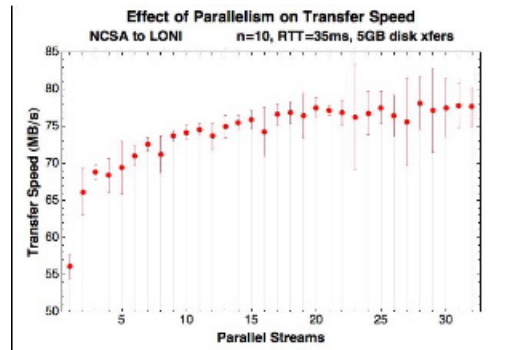
As a simple example, if I can put one byte per second onto the network, and it takes 10 seconds for that byte to get there, and 10 seconds for the acknowledgment to get back (RTT = 20 seconds), then I would need at least 20 bytes of storage. Then, hopefully, by the time I am ready to send byte 21, I 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-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 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 5.0.2.

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

Name

`globus-url-sync` -- Used in conjunction with `globus-url-copy` to synchronize directories.

`globus-url-sync`

Tool description

globus-url-sync is a command line tool which provides a list of files to be transferred, in order to synchronize two directories. It currently supports `gsiftp://` (GridFTP) and `sshftp://` protocol specifiers in the URL.

The program **globus-url-sync** compares two endpoints, using GridFTP, and prints a list of GSI file transfers that should be performed using **globus-url-copy**.

The current implementation of **globus-url-sync** supports very basic features for directory synchronization. It includes comparators for existence checks, file size checks, modification timestamp checks, but not checksum comparison.

- [Before you begin](#)
- [Command syntax](#)
- [Command line options](#)
- [Limitations](#)

Before you begin

1. First, as with **globus-url-copy**, you must have a valid proxy certificate to run **globus-url-sync** using protocol "`gsiftp://`".

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](#)¹, 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.

2. 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](#).

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

Command syntax

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

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

where:

[optional command line switches]	See Command line options 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 that need to be synchronized will be listed.
<i>Destination_URL</i>	Specifies the URL where you want to copy the files. The types of the source and the destination must match. In other words, if the source is a file, the destination must be a file, and if the source is a directory, the destination must be a directory.

Note

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

URL prefixes

The following URL prefixes are supported:

- **gsiftp://**
- **sshftp://**

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 that are not synchronized will be listed.

Table 2. URL formats

<code>gsiftp://myhost.mydomain.com//tmp/file1</code>	File name, absolute path.
<code>gsiftp://myhost.mydomain.com/~file1</code>	File name, absolute path.
<code>gsiftp://myhost.mydomain.com/file1</code>	File name, relative path.
<code>gsiftp://myhost.mydomain.com//tmp/dir1/</code>	Directory, absolute path.

Command line options

-help -usage	Print help text.
-version	Print the version of this program.
-d -debug -v -verbose	Print additional detail.
-r -recursive-dir-copy	Output directory names, when an entire directory is to be copied recursively.
-n -newer	File is to be transferred, if the source timestamp is newer than the destination timestamp.
-o -older	File is to be transferred, if the source timestamp is older than the destination timestamp.
-s -size	File is to be transferred, if the sizes of the source and the destination are not the same.

Limitations

- This is an early version of **globus-url-sync**. In the event that unexpected results are returned, please re-run the command with the **-verbose** option.
- **globus-url-copy** should be invoked with the **-r** (copy files in subdirectories) **-cd** (create directory) options, so that directories can be copied recursively (for "globus-url-sync -r"), and so that directories at the destination can be created.
- Authentication errors may be erroneously be reported as though a file is missing.
- Order of options does not currently effect order in which matching criteria are evaluated.

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 all 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 Section 4.4.1, “Running in daemon mode” for more information. Default value: TRUE
detach <0 1>, -S, -detach	Run as a background daemon detached from any controlling terminals. See Section 4.4.1, “Running in daemon mode” for more information. Default value: FALSE
ssh, -ssh	Run over a connected ssh session. Default value: not set

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 <u>daemon mode</u> . Default value: not set
chdir <0 1>, -chdir	Change directory when the server starts. This will change directory to the dir specified by the chdir_to option. Default value: TRUE
chdir_to <string>, -chdir-to <string>	Directory to chdir to after starting. Will use / if not set. Default value: not set
fork <0 1>, -f, -fork	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
single <0 1>, -1, -single	Exit after a single connection. Default value: FALSE

Authentication, Authorization, and Security Options

auth_level <number>, -auth-level <number>	<ul style="list-style-type: none"> • 0 = Disables all authorization checks. • 1 = Authorize identity only. • 2 = Authorize all file/resource accesses. <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>
ipc_allow_from <string>, -ipc-allow-from <string>	<p>Only allow IPC connections (applicable for backend servers in a striped configuration) 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.</p> <p>Default value: not set</p>
ipc_deny_from <string>, -ipc-deny-from <string>	<p>Deny IPC connections (applicable for backend servers in a striped configuration) 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.</p> <p>Default value: not set</p>
allow_from <string>, -allow-from <string>	<p>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.</p>

	Default value: not set
deny_from <string> , -deny-from <string>	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
secure_ipc <0 1> , -si, -secure-ipc	Use GSI security on the IPC channel. Default value: TRUE
ipc_auth_mode <string> , -ia <string> , -ipc-auth- mode <string>	Set GSI authorization mode for the IPC connection. Options are one of the following: <ul style="list-style-type: none"> • none • host • self • subject:[subject] Default value: host
allow_anonym- ous <0 1> , -aa , -allow- anonymous	Allow cleartext anonymous access. If server is running as root, anonymous_user must also be set. Disables IPC security. Default value: FALSE
anonym- ous_names_al- lowed <string> , -an- onymous- names-allowed <string>	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
anonym- ous_user <string> , -an- onymous-user <string>	User to setuid to for an anonymous connection. Only applies when running as root. Default value: not set
anonym- ous_group <string> , -an- onymous-group <string>	Group to setgid to for an anonymous connection. If not set, the default group of anonymous_user will be used. Default value: not set
pw_file <string> , -password- file <string>	Enable cleartext access and authenticate users against this /etc/passwd formatted file. Default value: not set
connec- tions_max	Maximum concurrent connections allowed. Only applies when running in <u>daemon mode</u> . Unlimited if not set.

<code><number> , -connections- max <number></code>	Default value: not set
<code>connec- tions_dis- abled <0 1> , -connections- disabled</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
<code>offline_msg <string> , -offline-msg <string></code>	Custom message to be displayed to clients when the server is offline via the <code>connections_disabled</code> or <code>connections_max = 0</code> options. Default value: not set
<code>disable_com- mand_list <string> , -disable-com- mand-list <string></code>	A comma separated list of client commands that will be disabled. Default value: not set
<code>authz_cal- louts , -cas , -authz-cal- louts</code>	Enable the GSI authorization callout framework, for callouts such as CAS. Default value: TRUE
<code>acl , -em , -acl</code>	A comma separated list of ACL or event modules to load. Default value: not set

Logging Options

<code>log_level <string> , -d <string> , -log-level <string></code>	Log level. A comma-separated list of levels from the following: <ul style="list-style-type: none"> • ERROR • WARN • INFO • DUMP • ALL For example: <pre>globus-gridftp-server -d error,warn,info</pre> You may also specify a numeric level of 1-255. Default value: ERROR
---	---

<code>log_module <string> ,</code>	Indicates the <code>globus_logging</code> module that will be loaded. If not set, the default <code>stdio</code> module will be used and the logfile options (see next option) will apply.
--	--

- `-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 <string>, -l <string>, -logfile <string>` 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.
- Default value: not set
- `log_unique <string>, -L <string>, -logdir <string>` Partial path to which `gridftp.(pid).log` will be appended to construct the log filename.
- Example:
- ```
-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.EVT=FTP_INFO START=20050520163008.305913 USER=ftp FILE=/etc/group BUFFER=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).
- `DATE`: time the transfer completed.
 - `START`: time the transfer started.
 - `HOST`: hostname of the server.
 - `USER`: username on the host that transferred the file.
 - `BUFFER`: tcp buffer size (if 0 system defaults were used).
 - `BLOCK`: the size of the data block read from the disk and posted to the network.
 - `NBYTES`: the total number of bytes transferred.

- VOLUME: the disk partition where the transfer file is stored.
- STREAMS: the number of parallel TCP streams used in the transfer.
- STRIPES: the number of stripes used on this end of the transfer.
- DEST: the destination host.
- TYPE: the transfer type, RETR is a send and STOR is a receive (ftp 959 commands).
- CODE: the FTP rfc959 completion code of the transfer. 226 indicates success, 5xx or 4xx are failure codes.

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_usage_stats <0|1> , -disable-usage-stats

Disable transmission of per-transfer usage statistics. See the [Usage Statistics](#)¹ section in the online documentation for more information.

Default value: FALSE

usage_stats_target <string> ,
-usage-stats-target <string>

Comma-separated list of contact strings for usage statistics listeners. The format of <string> is host:port.

Default value: usage-stats.globus.org:4810

Example:

```
-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).

The usage stats sent to a particular receiver may be customized by configuring it with a taglist (host:port!taglist) The taglist is a list of characters that each correspond to a usage stats tag. When this option is unset, stats are reported to usage-stats.globus.org:4810. If you set your own receiver, and wish to continue reporting to the Globus receiver, you will need to add it manually. The list of available tags follow. Tags marked * are reported by default.

- *(e) START - start time of transfer
- *(E) END - end time of transfer
- *(v) VER - version string of gridftp server
- *(b) BUFFER - tcp buffer size used for transfer
- *(B) BLOCK - disk blocksize used for transfer
- *(N) NBYTES - number of bytes transferred

¹ ../../Usage_Stats.html

- *(s) STREAMS - number of parallel streams used
- *(S) STRIPES - number of stripes used
- *(t) TYPE - transfer command: RETR, STOR, LIST, etc
- *(c) CODE - ftp result code (226 = success, 5xx = fail)
- *(D) DSI - DSI module in use
- *(A) EM - event modules in use
- *(T) SCHEME - ftp, gsiftp, sshftp, etc. (client supplied)
- *(a) APP - guc, rft, generic library app, etc. (client supplied)
- *(V) APPVER - version string of above. (client supplied)
- (f) FILE - name of file/data transferred
- (i) CLIENTIP - ip address of host running client (control channel)
- (I) DATAIP - ip address of source/dest host of data (data channel)
- (u) USER - local user name the transfer was performed as
- (d) USERDN - DN that was mapped to user id
- (C) CONFID - ID defined by -usage-stats-id config option
- (U) SESSID - unique id that can be used to match transfers in a session and transfers across source/dest of a third party transfer. (client supplied)

us- Identifying tag to include in usage statistics data.

age_stats_id
 <string>, -us- Default value: not set
 age-stats-id
 <string>

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>, -re-
 mote-nodes Default value: not set
 <string>

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_blocks- Size in bytes of sequential data that each stripe will transfer.
 ize <number>,
 -sbs <number> Default value: 1048576
 , -stripe-

blocksize

<number>

stripe_count Number of stripes to use per transfer when this server controls that number. If remote nodes are statically configured (via -r or remote_nodes), this will be set to that number of nodes, otherwise the default is 1.

<number> ,
-stripe-count
<number>

Default value: not set

stripe_layout Stripe layout. 1 = Partitioned, 2 = Blocked.

<number> , -sl
<number> ,
-stripe-lay-
out <number>

Default value: 2

stripe_blocksize_locked Do not allow client to override stripe blocksize with the **OPTS RETR** command.

<0|1> ,
-stripe-block-
size-locked;

Default value: FALSE

stripe_layout_locked Do not allow client to override stripe layout with the **OPTS RETR** command.

<0|1> ,
-stripe-lay-
out-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 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.

<0|1> , -sync-
writes

Default value: FALSE

use_home_dirs Set the startup directory to the authenticated users home dir.

, -use-home-
dirs

Default value: TRUE

perms Set the default permissions for created files. Should be an octal number such as 0644. The default is 0644. Note: If umask is set it will affect this setting -- i.e. if the umask is 0002 and this setting is 0666, the resulting files will be created with permissions of 0664.

<string> ,
-perms
<string>

Default value: not set

file_timeout Timeout in seconds for all disk accesses. A value of 0 disables the timeout.

<number> ,

Default value: not set

-file-timeout
<number>

Network Options

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

, -control-interface
<string> Default value: not set

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

, -data-interface
<string> Default value: not set

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

-ipc-interface
<string> Default value: not set

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

-hostname
<string> Default value: not set

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

-ipc-port
<number> Default value: not set

Timeouts

control_preauth_timeout Time in seconds to allow a client to remain connected to the control channel without activity before authenticating.

<number>, Default value: 120
-control-preauth-timeout
<number>

control_idle_timeout Time in seconds to allow a client to remain connected to the control channel without activity.

<number>;, Default value: 600
-control-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

banner_append When this is set, the message set in the 'banner' or 'banner_file' option will be appended to the default banner message rather than replacing it.
 <0|1> , -ban-
 ner-append Default value: FALSE

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

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

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
- dc_whitelist <string> , -dc-whitelist <string> A comma separated list of drivers allowed on the network stack.
 Default value: not set
- fs_whitelist <string> , -fs-whitelist <string> A comma separated list of drivers allowed on the disk stack.
 Default value: not set
- popen_whitelist <string> , -popen-whitelist <string> A comma separated list of programs that the popen driver is allowed to execute, when used on the network or disk stack. An alias may also be specified, so that a client does not need to specify the full path. Format is [alias:]prog,[alias:]prog. example: /bin/gzip,tar:/bin/tar
 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
- 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

1. Globus GridFTP GUI (pre-alpha)

The Globus GridFTP GUI is Java web start application. Users can get it by clicking a link; the program will be downloaded and started automatically. A pre-alpha version of the GUI is available now.

- [Download the GUI client](#)¹

The GUI client provides an easy-to-use interface for connecting to GridFTP servers and transferring files. It has the following features:

- Allows you to browse the local file system and transfer files and directories between the local system and remote GridFTP servers and between two remote GridFTP servers (third-party transfers).
- Supports file system operations such as creating, deleting and renaming files and directories.

Prerequisites:

- JDK 1.5.0+

Supported Platforms:

- Windows
- Linux
- MAC

The GUI provides two ways for generating a proxy credential required for the data transfer:

1. Creating a proxy credential using a locally stored key pair.
2. Obtaining a proxy from a MyProxy Server. For more information about MyProxy, please visit: <http://myproxy.ncsa.uiuc.edu/>.

[A demo of using the GridFTP GUI is available here](#)². Open the file ending in .htm with any browser with the Flash plugin to start the Flash demo - then just click the green arrows to progress through each screen.

2. UberFTP

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.

¹ <http://www-unix.globus.org/cog/demo/ogce/ftp.jnlp>

² [../demo.tar.gz](#)

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. Typical configuration

The following describes a typical GridFTP configuration of the front end (control channel) and back end (data channels). For other alternatives that provide greater levels of security, see [Advanced Configuration](#).

By default, 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. This configuration is also good for testing purposes.

3. Firewall requirements

If the GridFTP server is behind a firewall:

1. Contact your network administrator to open up port 2811 (for GridFTP control channel connection) and a range of ports (for GridFTP data channel connections) for the incoming connections. If the firewall blocks the outgoing connections, open up a range of ports for outgoing connections as well.
2. Set the environment variable `GLOBUS_TCP_PORT_RANGE`:

```
export GLOBUS_TCP_PORT_RANGE=min,max
```

where `min,max` specify the port range that you have opened for the incoming connections on the firewall. This restricts the listening ports of the GridFTP server to this range. Recommended range is 1000 (e.g., 50000-51000) but it really depends on how much use you expect.

3. If you have a firewall blocking the outgoing connections and you have opened a range of ports, set the environment variable `GLOBUS_TCP_SOURCE_RANGE`:

```
export GLOBUS_TCP_SOURCE_RANGE=min,max
```

where `min,max` specify the port range that you have opened for the outgoing connections on the firewall. This restricts the outbound ports of the GridFTP server to this range. Recommended range is twice the range used for `GLOBUS_TCP_PORT_RANGE`, because if parallel TCP streams are used for transfers, the listening port would remain the same for each connection but the connecting port would be different for each connection.



Note

If the server is behind NAT, the `--data-interface <real ip/hostname>` option needs to be used on the server.

If the GridFTP *client* is behind a firewall:

1. Contact your network administrator to open up a range of ports (for GridFTP data channel connections) for the incoming connections. If the firewall blocks the outgoing connections, open up a range of ports for outgoing connections as well.
2. Set the environment variable `GLOBUS_TCP_PORT_RANGE`

```
export GLOBUS_TCP_PORT_RANGE=min,max
```

where `min,max` specify the port range that you have opened for the incoming connections on the firewall. This restricts the listening ports of the GridFTP client to this range. Recommended range is 1000 (e.g., 50000-51000) but it really depends on how much use you expect.

3. If you have a firewall blocking the outgoing connections and you have opened a range of ports, set the environment variable `GLOBUS_TCP_SOURCE_RANGE`:

```
export GLOBUS_TCP_SOURCE_RANGE=min,max
```

where `min,max` specify the port range that you have opened for the outgoing connections on the firewall. This restricts the outbound ports of the GridFTP client to this range. Recommended range is twice the range used for `GLOBUS_TCP_PORT_RANGE`, because if parallel TCP streams are used for transfers, the listening port would remain the same for each connection but the connecting port would be different for each connection.

Additional information on Globus Toolkit Firewall Requirements is available [here](#)¹.

4. Configuring Security for GridFTP

There are many security options in GridFTP ranging from no security to higher security via GSI .

4.1. Anonymous mode

Anonymous mode (using the `-aa` option) allows any user with an FTP client to read and write (and delete) files that the server process can similarly access (it is also a quick way to test that your server works).

```
% globus-gridftp-server -aa
    Server listening at 127.0.0.1:58806
```

Warning

When the server is run in this way, anyone who can connect to the server will possess all the same rights as the user that the process is run as (directly or via `-anonymous-user`). If using this mode intentionally for open access, it is best to run under a dedicated account with limited filesystem permissions. You can also use the option below to disable FTP commands such as STOR, ESTO, DELE, RDEL, RNT0, etc to make sure that users can only read from the server and not write to it.

```
-disable-command-list <string>
```

Where `<string>` represents a comma separated list of client commands that will be disabled. Default: not set.

4.2. Username/password

If you trust your network and want a minimal amount of security, you can run the `globus-gridftp-server` with clear text passwords. This security model is the one originally introduced in RFC959.

Warning

We do not recommend it for long running servers open to the internet.

4.2.1. Create password file

To run the server in clear text password mode, we first need to create a password file dedicated to it. The format of the password file is the same as standard system password files; however, it is ill-advised to use a system password file. To create an entry in a GridFTP password file, run the following commands:

```
% touch pwfile
    % gridftp-password.pl >> pwfile
    Password:
```

This will ask you for a password and then create an entry in the password file for the current user name and the given password. Take a look at the file created. You will notice that the password you typed in is not in the file in a clear text form. We have run it through a one way hash algorithm before storing it in the file.

¹ <http://www.globus.org/toolkit/security/firewalls/>

4.2.2. Run the server in password mode

Simply start the server pointing it at the password file you just created.

```
% globus-gridftp-server -password-file /full/path/of/pwfile
    Server listening at 127.0.0.1:5555
```

4.2.3. Make a transfer

To run `globus-url-copy` with the password, use the following syntax:

```
globus-url-copy file:///etc/group ftp://username:pw@localhost:5000/tmp/group
```

4.3. SSHFTP (GridFTP-over-SSH)

This type of security introduces the `sshftp` control channel (frontend) protocol. This is a very simple means of obtaining strong security on the control channel only (the data channel is *not* authenticated). With this approach, you can run a GridFTP transfer anywhere that you can `ssh`. `sshftp://` leverages the ubiquitous `ssh/ssh` programs to form control channel connections much in the same way that `inetd` forms connections.

4.3.1. Configure Client-Side `sshftp://`

Every `$GLOBUS_LOCATION` must be configured for client-side `sshftp://` connections. In other words, if we wish to use `globus-url-copy` with `sshftp://` URLs, we must first configure the `$GLOBUS_LOCATION` that contains `globus-url-copy` in the following way:

```
% $GLOBUS_LOCATION/setup/globus/setup-globus-gridftp-sshftp
```

4.3.2. Configure Server Side `sshftp://`

Every host that wishes to run a `globus-gridftp-server` which can accept `sshftp://` connections must run the following command as root:

```
% $GLOBUS_LOCATION/setup/globus/setup-globus-gridftp-sshftp -server
```

In the absence of root access, a user can configure the server to allow `sshftp://` connections for that user only with the following command:

```
% $GLOBUS_LOCATION/setup/globus/setup-globus-gridftp-sshftp -server -nonroot
```

The above command creates a file named `'sshftp'` in `'/etc/grid-security'` (if run as root) or in `'$HOME/.globus'` (if run as nonroot). The default contents of the `'sshftp'` file is shown below. To configure the GridFTP server for `sshftp` transfers, you have to edit this file.

```
export GLOBUS_LOCATION=/sandbox/kettimut/421/INSTALL
. $GLOBUS_LOCATION/etc/globus-user-env.sh

#export GLOBUS_TCP_PORT_RANGE=50000,50100

$GLOBUS_LOCATION/sbin/globus-gridftp-server -ssh
# -data-interface <interface to force data connections>
```

4.3.3. Performing `sshftp://` Transfers

In this case, a `globus-gridftp-server` does not need to be running. The server will be started via the `sshd` program. Therefore, the hostname and port should be that of the `sshd` server. Run `globus-url-copy` just as you have before; simply change `ftp://` to `sshftp://`.

```
% globus-url-copy -v file:/etc/group sshftp://127.0.0.1/tmp/group % globus-url-copy -list
```

4.4. GSIFTP

This security option can be the most involved to set up, but provides the most security. It requires setting up GSI security as described in the GT Installation Guide here: [Basic Security Configuration](#).

Once GSI has been set up (host and user credentials are valid, the gridmap file is updated and you've run `grid-proxy-init` to create a proxy certificate), you simply run the GridFTP server:

```
globus-gridftp-server
```



Note

If run as `root`, it will pick up the host cert; if not, it will pick up the user cert.

Now you are ready to perform a GSI-authenticated transfer:

```
globus-url-copy <-s subject> src_url dst_url
```



Note

The `subject` option is only needed if the server was not started as `root`.

4.4.1. 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`.

4.4.2. Running under inetd or xinetd

Note

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

4.4.2.1. 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`.

Note

You should NOT include `USERID` in the log lines. See [Section 5, “High latency for GridFTP server connections”](#) for more information.

4.4.2.2. globus-gridftp-server -i

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

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

4.5. User permissions

Users are mapped to a local account on the server machine and file permissions are handled by the operating systems. In the anonymous mode, users that connect to the server will possess all the same rights as the user that the server process is run as (directly or via `-anonymous-user`).

In case of username/password authentication, the users are mapped to the uid corresponding to the username in the GridFTP password file and the access permissions for the users is same as that of the UID that they are mapped to. If SSH based authentication is used, upon successful authentication, SSHD maps users to a local account and the GridFTP server is run as the mapped local user. The access permissions are the same as that of the mapped local user.

If GSI is used, upon successful authentication an authorization callout is invoked to (a) verify authorization and (b) determine the local user id as which the request should be executed. This callout is linked dynamically. Globus GridFTP provides an implementation that supports a Globus "gridmapfile". Sites can also provide alternative implementations. Server does a setuid to the local user id as determined by the authorization callout and the access permissions are the same as that of the local user id.

GridFTP server provides an option to disable certain FTP commands:

```
-disable-command-list <string>
```

Where `<string>` represents a comma separated list of client commands that will be disabled. Default: not set.

5. 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 3, Key Admin Settings and Tuning Recommendations](#).

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.

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>

Error Code	Definition	Possible Solutions
530-globus_xio: Authentication Error 530-globus_gsi_callback_module: Could not verify credential 530-globus_gsi_callback_module: Could not verify credential 530-globus_gsi_callback_module: Invalid CRL: The available CRL has expired 530 End.	This error message indicates one of the following: Certificate Revocation List (CRL) for the source or destination server CA at the client has expired or CRL for client CA has expired at source or destination server or CRL for source (destination) server CA has expired at destination (source) server. CRL is a file {CA_hash}.r0 in /etc/grid-security/certificates or \${USER_HOME}/.globus/certificates or \${X509_CERT_DIR}	The tool available at http://dist.eu-gridpma.info/distribution/util/fetch-crl/ can be run in a crontab to keep the CRLs up to date.

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](#).