GT 5.2.4 Component Guide to Public Interfaces: GridFTP
GT 5.2.4 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-gridftp-server ....................................................................................................... 17

2. Graphical User Interface ................................................................................................... 30
   1. Globus Online .............................................................................................................. 30

3. Configuring GridFTP ......................................................................................................... 31
   1. GridFTP server configuration overview ....................................................................... 31
   2. Typical configuration ................................................................................................. 31
   3. Firewall requirements ............................................................................................... 31
   4. Configuring Security for GridFTP ............................................................................... 33
   5. **globus-gridftp-server** quickstart ........................................................................... 38

4. Environment variable interface ....................................................................................... 39
   1. Environment variables for GridFTP .............................................................................. 39

A. Errors ............................................................................................................................... 40

Glossary .................................................................................................................................. 42
List of Figures

1. Effect of Parallel Streams in GridFTP ................................................................. 15
List of Tables

1. URL formats ................................................................................................................................. 6
A.1. GridFTP Errors ....................................................................................................................... 40
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" 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" 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" 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[^1]**
- **C Control Library API[^1]**

For information on the internationalization API, see Chapter 1.
GridFTP Commands
**Name**
globus-url-copy — Multi-protocol data movement

**Synopsis**
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 CA¹, 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.

3. 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:

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

**Note**

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

## URL prefixes

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

¹ [http://www.doegrids.org/pages/cert-request.htm](http://www.doegrids.org/pages/cert-request.htm)
• file:// (on a local machine only)
• ftp://
• gsiftp://
• http://
• 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`/`-re` curse switch described below.

**Table 1. URL formats**

<table>
<thead>
<tr>
<th>URL Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://myhost.mydomain.com/myweb-page/default.html">http://myhost.mydomain.com/myweb-page/default.html</a></td>
<td>Port is not specified; therefore, GridFTP uses protocol default (in this case, 80).</td>
</tr>
<tr>
<td>file:///foo.dat</td>
<td>Host is not specified; therefore, GridFTP uses your local host. Port is not specified; therefore, GridFTP uses protocol default (in this case, 80).</td>
</tr>
<tr>
<td>file://foo.dat</td>
<td>This is also valid but is not recommended because, while many servers (including ours) accept this format, it is <strong>not</strong> RFC conformant and is not recommended.</td>
</tr>
</tbody>
</table>

**Important**

For GridFTP (gsiftp://) and FTP (ftp://), it is legal to specify a user name and password in 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 distin-
guished 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:
  - number of bytes transferred,
  - performance since the last update (currently every 5 seconds), and
  - average performance for the whole transfer.
- **-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.
- **-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 detection⁴.

**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 filename

Reads a list of URL pairs from a filename.

Each line should contain:

sourceURL destURL

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.

-rp | -relative-paths

The path portion of ftp urls will be interpereted as relative to the user’s starting directory on the server. By default, all paths are root-relative. When this flag is set, the path portion of the ftp url must start with %2F if it designates a root-relative path.

-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.

-do <filename> | -dump-only <filename>

Perform no write operations on the destination. Instead, all files that would be transferred are enumerated and dumped to the specified file. Resulting file is the same format as the -f input file. Note: if you intend to use this file as input
for a future transfer, the -create-dest option will be required if any destination directories do not already exist.

-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

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.

**Note**

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 to enable threading in the
client. To switch to threaded flavor, set the environment variable
'GLOBUS_THREAD_MODEL=pthread'.

-fast

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

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> Sets the source end of a remote transfer to use piped-in input with the given command line.

⚠️ Warning
Do not use with the -fsstack option.

-dst-pipe | -DP <command line> Sets the destination end of a remote transfer to write data to then standard input of the program run via the given command line.

⚠️ Warning
Do not use with the -fsstack option.

-pipe <command line> Sets both -src-pipe and -dst-pipe to the same value.

destack | -data-channel-stack Specifies the XIO driver stack for the network on both the source and 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-dstack | -source-data-channel-stack Specifies the XIO driver stack for the network on the source GridFTP server.

-src-fstack | -source-file-system-stack Specifies the XIO driver stack for the disk on the source GridFTP server.

-dst-dstack | -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` or `-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):

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

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

\[-tcp-bs = 100 \times 50 \times 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.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 5.2.4.

**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](http://dims.ncsa.uiuc.edu/set/uberftp/download.html)
Name
globus-gridftp-server — Configures the GridFTP Server

Synopsis
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 \texttt{-c <configfile>} command line option.
- \$GLOBUS\_LOCATION/etc/gridftp.conf
- /etc/grid-security/gridftp.conf

Options are one per line, with the format:

\texttt{<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:

\begin{verbatim}
port 5000
allow_anonymous 1
anonymous_user bob
banner "Welcome!"
\end{verbatim}

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'.

**Informational Options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>help `&lt;0</td>
<td>1&gt;<code>, </code>-h<code>, </code>-help`</td>
<td>Show usage information and exit.</td>
</tr>
<tr>
<td>version `&lt;0</td>
<td>1&gt;<code>, </code>-v<code>, </code>-version`</td>
<td>Show version information for the server and exit.</td>
</tr>
<tr>
<td>versions `&lt;0</td>
<td>1&gt;<code>, </code>-V<code>, </code>-versions`</td>
<td>Show version information for all loaded globus libraries and exit.</td>
</tr>
</tbody>
</table>

**Modes of Operation**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>inetd `&lt;0</td>
<td>1&gt;<code>, </code>-i<code>, </code>-inetd`</td>
<td>Run under an inetd service.</td>
</tr>
<tr>
<td>daemon `&lt;0</td>
<td>1&gt;<code>, </code>-s<code>, </code>-daemon`</td>
<td>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.</td>
</tr>
<tr>
<td>detach `&lt;0</td>
<td>1&gt;<code>, </code>-S<code>, </code>-detach`</td>
<td>Run as a background daemon detached from any controlling terminals. See Section 4.4.1, “Running in daemon mode” for more information.</td>
</tr>
<tr>
<td>ssh, <code>-ssh</code></td>
<td>Run over a connected ssh session.</td>
<td>not set</td>
</tr>
<tr>
<td>exec <code>&lt;string&gt;</code>, <code>-exec &lt;string&gt;</code></td>
<td>For statically compiled or non-GLOBUS_LOCATION standard binary locations, specify the full path of the server binary here. Only needed when run in daemon mode.</td>
<td></td>
</tr>
</tbody>
</table>
globus-gridftp-server

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: not set

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

chroot_path <string>, -chroot-path <string>

Path to become the new root after authentication. This path must contain a valid certificate structure, /etc/passwd, and /etc/groups. The command globus-gridftp-server-setup-chroot can help create a suitable directory structure.

Default value: not set

Authenticaton, Authorization, and Security Options

auth_level <number>, -auth-level <number>

- 0 = Disables all authorization checks.
- 1 = Authorize identity only.
- 2 = Authorize all file/resource accesses.

If not set, the GridFTP Server uses level 2 for front ends and level 1 for data nodes.

Default value: not set

ipc_allow_from <string>, -ipc-allow-from <string>

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.

Default value: not set

ipc_deny_from <string>, -ipc-deny-from <string>

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.
globus-gridftp-server

allow_from <string>, -allow-from <string>

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

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:

• none
• host
• self
• subject:[subject]

Default value: host

allow_anonymous <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

anonymous_names_allowed <string>, -anonymous-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 username.

Default value: not set

anonymous_user <string>, -anonymous-user <string>

User to setuid to for an anonymous connection. Only applies when running as root.

Default value: not set

anonymous_group <string>, -anonymous-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

allow_root, -allow-root

Allow clients to be mapped to the root account.

Default value: FALSE
pw_file <string>, -password-file <string>

Enable cleartext access and authenticate users against this /etc/passwd formatted file.

Default value: not set

connections_max <number>, -connections-max <number>

Maximum concurrent connections allowed. Only applies when running in daemon mode. Unlimited if not set.

Default value: not set

connections_disabled <0|1>, -connections-disabled

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

offline_msg <string>, -offline-msg <string>

Custom message to be displayed to clients when the server is offline via the connections_disabled or connections_max = 0 options.

Default value: not set

disable_command_list <string>, -disable-command-list <string>

A comma separated list of client commands that will be disabled.

Default value: not set

authz_callouts , -authz-callouts

Enable the GSI authorization callout framework.

Default value: TRUE

restrict_paths , -rp , -restrict-paths

A comma separated list of full paths that clients may access. Each path may be prefixed by R and/or W, denoting read or write access, otherwise full access is granted. If a given path is a directory, all contents and subdirectories will be given the same access. Order of paths does not matter -- the permissions on the longest matching path will apply. The special character ‘~’ will be replaced by the authenticated user's home directory. Note that if the authenticated user's home directory is not accessible, the home directory and starting path will be set to '/'. By default all paths are allowed, and access control is handled by the OS.

Default value: not set

rp_follow_symlinks , -rp-follow-symlinks

Allow following symlinks that lead to restricted paths.

Default value: FALSE

acl , -em, -acl

A comma separated list of ACL or event modules to load.

Default value: not set

Logging Options

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

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

- 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 <string>, -log-module <string>**

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.

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.

**Note**

You have to provide full path

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.

**Note**

You have to provide full path
log_transfer <string>, -Z <string>, -log-transfer <string>

Log NetLogger-style info for each transfer into this file.

**Note**

You have to provide full path

Example: DATE=20050520163008.305913 HOST=localhost PROG=globus-gridftp-server NL_EVNT=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 YYYYMMDDHHMMS. UUUUUU (microsecs).

- **DATE**: time the transfer completed.
- **START**: time the transfer started.
- **HOST**: hostname of the server.
- **USER**: username on the host that transfered 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 section in the online documentation for more information.
Default value: FALSE

Usage statistics target list:

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:**

```bash
```

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
- *(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)

usage_stats_id <string>, -usage-stats-id <string>
Identifying tag to include in usage statistics data.
Default value: not set

**Single and Striped Remote Data Node Options**

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

data_node <0|1>, -dn, -data-node
This server is a back end data node. See Separation of processes for higher security for an example of using this option.
Default value: FALSE

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

stripe_count <number>, -stripe-count <number>
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.
Default value: not set

stripe_layout <number>, -sl <number>, -stripe-layout <number>
Stripe layout. 1 = Partitioned, 2 = Blocked.
Default value: 2

stripe_blocksize_locked <0|1>, -stripe-blocksize-locked;
Do not allow client to override stripe blocksize with the OPTS RETR command.
Default value: FALSE

stripe_layout_locked <0|1>, -stripe-layout-locked
Do not allow client to override stripe layout with the OPTS RETR command.
Default value: FALSE

**Disk Options**

blocksize <number>, -bs <number>, -blocksize <number>
Size in bytes of data blocks to read from disk before posting to the network.
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

### use_home_dirs , -use-home-dirs

Set the startup directory to the authenticated user's home dir.

Default value: TRUE

### perms <string>, -perms <string>

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.

Default value: not set

### file_timeout <number>, -file-timeout <number>

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

Default value: not set

### 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
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: 120

control_idle_timeout <number>, -control-idle-timeout <number>
Time in seconds to allow a client to remain connected to the control channel without activity.
Default value: 600

ipc_idle_timeout <number>, -ipc-idle-timeout <number>
Idle time in seconds before an unused IPC connection will close.
Default value: 900

ipc_connect_timeout <number>, -ipc-connect-timeout <number>
Time in seconds before cancelling an attempted IPC connection.
Default value: 60

port_range <string>, -port-range <string>
Port range to use for incoming connections. The format is "startport,endport". This, along with -data-interface, can be used to enable operation behind a firewall and/or when NAT is involved. This is the same as setting the environment variable GLOBUS_TCP_PORT_RANGE.
Default value: not set

**User Messages**

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

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

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

banner_append <0|1>, -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.
Default value: FALSE

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

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

**Module Options**

load_dsi_module <string>, -dsi <string>
Load this Data Storage Interface module. File and remote modules are defined by the server. If not set, the file module is loaded, unless the remote option is specified, in which case the remote module is loaded. An additional config-
uration 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>, -allowed-modules <string>**
Comma-separated list of ERET/ESTO modules to allow and, optionally, specify an alias for. Example:
-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

**dc_default <string>, -dc-default <string>**
A comma separated list of XIO drivers and options representing the default network stack. Format is of each driver entry is driver1[:opt1=val1;opt2=val2;...]. The bottom of the stack, the transport driver, is always first.

Default value: not set

**fs_default <string>, -fs-default <string>**
A comma separated list of XIO drivers and options representing the default disk stack. Format is of each driver entry is driver1[:opt1=val1;opt2=val2;...]. The bottom of the stack, the transport driver, is always first.

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.

⚠️ **Note**

You have to provide full path

Default value: not set

**config_dir <string>, -C <string>**
Path to directory holding configuration files that should be loaded. Files will be loaded in alphabetical order, and in the event of duplicate parameters the
last loaded file will take precedence. Note that the main configuration file, if one exists, will always be loaded last.

Default value: not set

```
config_base_path <string>, -config-base-path <string>
```

Base path to use when config and log path options are not full paths. By default this is the current directory when the process is started.

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

```
pidfile <string>, -pid-file <string>
```

Write PID of the GridFTP server to this path. May contain variable references to `${localstatedir}`

Default value: not set

⚠️ **Warning**

Any FLAG can be negated by prepending `-no` or `-n` to the command line option.

**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](http://www.ogf.org/documents/GFD.20.pdf).

Globus GridFTP server does not run on windows
Chapter 2. Graphical User Interface

1. Globus Online

Globus Toolkit does not provide a client with Graphical User Interface (GUI) but Globus Online\(^1\) provides a web GUI for GridFTP data movement. It has the following features:

- With a one-click Globus Connect\(^2\) (modified GridFTP server binary) installed on your local system, you can 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\(^3\). 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.

---

\(^1\) [http://www.globusonline.org](http://www.globusonline.org)

\(^2\) [https://www.globusonline.org/globus_connect/](https://www.globusonline.org/globus_connect/)

\(^3\) [../demo.tar.gz](../demo.tar.gz)
Chapter 3. Configuring GridFTP

1. GridFTP server configuration overview

The configuration interface for GridFTP is the admin tool, `globus-gridftp-server(8)`, 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(8)`.

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 GridFTP server to bind to a local port in this range for outbound connections. 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 (local) 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 GridFTP client to bind to a local port in this range for outbound connections. 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](http://www.globus.org/toolkit/security/firewalls/).
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 \texttt{-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).

\begin{verbatim}
globus@elephant% globus-gridftp-server -aa
Server listening at 127.0.0.1:58806
\end{verbatim}

\textbf{Warning}

When the server is run in this way, anyone who can connect to the server will posses all the same rights as the user that the process is run as (directly or via \texttt{-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, RNTO, etc to make sure that users can only read from the server and not write to it.

\begin{verbatim}
-disable-command-list <string>
\end{verbatim}

Where \texttt{<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.

\textbf{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:

\begin{verbatim}
globus@elephant% touch pwfile
globus@elephant% gridftp-password.pl >> pwfile
Password:
\end{verbatim}

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 though a one way hash algorithm before storing it in the file.

4.2.2. Run the server in password mode

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

\begin{verbatim}
globus@elephant% globus-gridftp-server -password-file /full/path/of/pwfile
Server listening at 127.0.0.1:5555
\end{verbatim}
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/sshd programs to form control channel connections much in the same way that inetd forms connections.

4.3.1. Configure Server for SSH GridFTP Support

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

```
$ globus-gridftp-server-enable-sshftp
```

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-gridftp-server-enable-sshftp -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). You may edit this file to set gridftp commandline options or environment variables such as GLOBUS_TCP_PORT_RANGE, but you can also set those options in the config file.

4.3.2. 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 ssdh 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 sshftp://127.0.0.1/tmp/
```

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
    }
```
Configuring GridFTP

```ini
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.

**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`

### 4.4.3. Running under launchd

launchd is used to start services in the system Mac OS X 10.4 (Tiger) and newer, and is also available as a port to FreeBSD.

Here is a sample configuration file for launchd to start the server:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
  <dict>
    <key>Disabled</key>
    <false/>
    <key>Label</key>
    <string>globus.gridftp</string>
    <key>Program</key>
    <string>(globus_location)/sbin/globus-gridftp-server</string>
    <key>ProgramArguments</key>
    <array>
      <string>globus-gridftp-server</string>
      <string>-i</string>
      <string>-d</string>
      <string>ALL</string>
      <string>-l</string>
      <string>(globus_location)/var/gridftp.log</string>
    </array>
  </dict>
</plist>
```

1 [http://www.globus.org/toolkit/docs/5.2/5.2.4/gridftp/globus-gridftp.plist](http://www.globus.org/toolkit/docs/5.2/5.2.4/gridftp/globus-gridftp.plist)
Provided that the configuration file is stored in `/System/Library/LaunchDae-
mons/globus.gridftp.plist`, the following command starts the GridFTP server:

```bash
launchctl load /System/Library/LaunchDaemons/globus.gridftp.plist
```

The server will be started automatically with a system, unless the value of the key 'Disabled' is changed from `<false/>` to `<true/>`.

### 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:

```
```

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](#).
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

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Definition</th>
<th>Possible Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>globus_ftp_client:</strong> the server responded with an error 530 530-</td>
<td>This error message indicates that the GridFTP server doesn't trust the certificate authority (CA) that issued your certificate.</td>
<td>You need to ask the GridFTP server administrator to install your CA certificate chain in the GridFTP server's trusted certificates directory.</td>
</tr>
<tr>
<td><strong>globus_xio:</strong> Authentication Error 530-OpenSSL Error: s3_srvr.c:2525: in library: SSL routines, function SSL3_GET_CLIENT_CERTIFICATE: no certificate returned 530-</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>globus_gsi_callback_module:</strong> Could not verify credential 530-</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>globus_gsi_callback_module:</strong> Can't get the local trusted CA certificate: Untrusted self-signed certificate in chain with hash d1b603c3 530 End.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>globus_ftp_control:</strong> gss_init_sec_context failed OpenSSL Error: s3_clnt.c:951: in library: SSL routines, function SSL3_GET_SERVER_CERTIFICATE: certificate verify failed</td>
<td>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.</td>
<td>You need to ask the resource administrator which CA issued their certificate and install the CA certificate in the local trusted certificates directory.</td>
</tr>
<tr>
<td><strong>globus_gsi_callback_module:</strong> Could not verify credential</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>globus_gsi_callback_module:</strong> Can't get the local trusted CA certificate: Untrusted self-signed certificate in chain with hash d1b603c3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>530-<strong>globus_xio:</strong> Authentication Error 530-</td>
<td>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</td>
<td>The tool available at <a href="http://dist.eugridpma.info/distribution/util/fetch-crl/">http://dist.eugridpma.info/distribution/util/fetch-crl/</a> can be run in a crontab to keep the CRLs up to date.</td>
</tr>
<tr>
<td><strong>globus_gsi_callback_module:</strong> Could not verify credential 530-</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>globus_gsi_callback_module:</strong> Could not verify</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

End.
<table>
<thead>
<tr>
<th>Error Code</th>
<th>Definition</th>
<th>Possible Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>fy credential 530-globus_gsi_callback_module: Invalid CRL: The available CRL has expired 530 End.</td>
<td>(destination) server CA has expired at destination (source) server. CRL is a file <code>{CA_hash}.r0</code> in <code>/etc/grid-security/certificates</code> or <code>${USER_HOME}/.globus/certificates</code> or <code>${X509_CERT_DIR}</code></td>
<td></td>
</tr>
</tbody>
</table>
# 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 component 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 supercomputer 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.