



GridFTP-Client

Arbeitspaket 3.2

Jan Ploski
OFFIS
Escherweg 2
26121 Oldenburg

Telefon: +49 441 9722-184
Telefax: +49 441 9722-102
E-Mail: jan.ploski@offis.de
Internet: <http://www.offis.de>

1 Einleitung

Das GridFTP-Protokoll unterstützt die effiziente Datenübertragung großer Datenmengen zwischen kompatiblen Servern. GridFTP-Server können entweder in Clustern installiert sein, die für die Ausführung von wissenschaftlichen Berechnungen zuständig sind, oder sie können den Zugriff auf dedizierte Speichereinheiten anbieten. Die letzteren können insbesondere für die mittelfristige Datensicherung gebraucht werden, wodurch es auch für kleinere wissenschaftlichen Einrichtungen möglich wird, große Datenmengen zu behalten, die für spätere Vergleiche oder Weiterverarbeitung benötigt werden.

Im D-Grid-Kontext wird vor allem die GridFTP-Implementierung von Globus Toolkit 4 als der Datenübermittlungsdienst in einzelnen Clustern eingesetzt. Vergleichsweise weniger bekannt, aber ebenso wichtig ist die GridFTP-Implementierung von dCache, der Middleware, die die dedizierten D-Grid-Speicherelemente in Jülich und Karlsruhe mit mehreren Terabytes an Bandkapazität zur Verfügung stellt.

Am Anfang von Arbeitspaket 3.2 (Standardisierte Datenspeicherung und Datenaustausch) stellten wir fest, dass überraschenderweise noch keine Client-Software existierte, die einen bequemen, interaktiven und zuverlässigen Zugang zu den beiden Arten von GridFTP-Servern bot. Angetrieben durch Anwenderanforderungen insb. seitens der Uni Oldenburg wurde in OFFIS ein grafischer GridFTP-Client entwickelt, der in seiner Benutzungsschnittstelle zwar einer gewöhnlichen FTP-Software ähnelt, aber zugleich die für große Datentransfers (z. B. mit >20000 Dateien, >500 GB Volumen) notwendigen Fehlertoleranzeigenschaften besitzt. Die Software wurde mit der dCache-Installation in Jülich getestet. Erfahrungen aus der Entwicklung wurden an dCache-Entwickler in Hamburg weitergereicht. Darüber hinaus wurde der GridFTP-Client an der Uni Oldenburg operativ (d. h. durch eigentliche Anwender) erfolgreich eingesetzt.

Die Weiterentwicklung des GridFTP-Clients wird als Open-Source-Projekt unter der Eclipse Public License erfolgen. Die erste Version, die in die populäre Entwicklungsumgebung Eclipse integriert ist, ist installationsbereit; eine von Eclipse unabhängige Version wird bis zum Arbeitspaketende veröffentlicht.

Das Softwareprodukt selbst ist unter der Adresse <http://bi.offis.de/gridftp/> veröffentlicht und steht damit ebenfalls Interessenten außerhalb des WISENT-Projekts zur Verfügung. Seit der Veröffentlichung Ende März 2008 verzeichnete diese Webseite über 300 verschiedene Besucher aus 33 Ländern, überwiegend aus Deutschland, USA und Großbritannien. Die interaktive Demonstration wurde dabei von über 70 Besuchern angeschaut, und die Software selbst über 40 mal heruntergeladen.

Der Rest dieses Dokuments, in englischer Sprache, besteht aus dem Benutzerhandbuch für den GridFTP-Client und schildert im Detail den Funktionsumfang der Software. Die aktuellste Version dieser Dokumentation wird bei der Installation geliefert bzw. kann separat heruntergeladen werden.

2 Introduction

The GridFTP protocol supports the efficient transfer of large amounts of data between conforming servers deployed as part of a (scientific) Grid infrastructure. These servers may be either a part of execution sites to which user jobs are submitted, or they may provide access dedicated storage elements with substantial disk or tape capacity for archiving Grid users' data. For example, the GridFTP server shipped in the popular Globus Toolkit middleware belongs to the former, and the GridFTP Door of the dCache middleware to the latter category.

During the course of the research project [WISENT](#), we found that, surprisingly, no user-friendly software existed that would provide convenient, reliable access to both types of GridFTP servers in order to browse, maintain and move around large (TeraByte scale) amounts of data accumulating in our project's research community. The tools provided by other projects were uncomfortable and restricted in functionality, not compatible with the German Grid's storage solution dCache, no longer maintained by their authors, and not at all integrated with the development environment (Eclipse) we endorsed for the participants of our project. For all these reasons, and taking advantage of the libraries provided by the Globus project, we embarked on the development of our own, general-purpose graphical GridFTP client software that would resemble in its user interface and features the familiar, off-the-shelf FTP clients, while still supporting the large (hence, failure-prone) transfers and the security infrastructure typical of a Grid computing environment. Ultimately, we wanted a GridFTP client that would make the conceptually simple task of moving data from A to B as free from technical considerations as possible for our users. Having developed the first useful working prototype, we released it under the open-source Eclipse Public License for everyone to enjoy and to contribute their feedback.

The remaining parts of this guide give an overview of features, describe the required installation steps, configuration, and functionality of the GridFTP Client. If you prefer a more lively introduction and learning-by-doing, we also provide an [online screencast](#) which walks you through a typical session with the program. All users are recommended to read to gain a deeper conceptual understanding of the core functionality.

3 Features Overview

Here is a short overview of GridFTP Client's features to help you determine whether the GridFTP Client fulfills your needs:

- Create Grid proxy and authenticate using a Globus-compliant X509 certificate
- Transfer files in third-party mode between two GridFTP servers
- Transfer directories recursively, possibly overwriting contents at the target location
- Recursively compute disk space usage on a GridFTP server
- Create and remove directories and files on a GridFTP server
- Initiate multiple concurrent transfers between different servers
- Browse GridFTP servers' contents while background transfers are underway
- Pause and resume transfers
- Report progress and errors for transfers
- Automatically persist transfers between application restarts
- Lightweight and self-contained package with few dependencies on other software
- Transfer using multiple parallel data channels per file

Here is a short overview of quite important features that are not yet implemented:

- Non-third-party file transfers (from client's local file system to a GridFTP server or the other way around)
- Renaming of files and directories
- Client-server architecture for managing transfers from multiple clients
- Firewall-friendliness (it's just as bad as any GridFTP client)

In addition to these features, the performance and fault tolerance of the GridFTP Client still have some room for improvement.

4 Installation

This chapter describes the required installation steps for the GridFTP Client.

4.1 Prerequisites

In the longer term, we plan to deliver the GridFTP Client in two variants:

- A plug-in variant integrated with the Eclipse development environment
- A standalone, easier to install "rich client" variant which doesn't need Eclipse

However, at present only the first variant is released, to which the following description applies.

4.1.1 Eclipse

Before installing the GridFTP Client plug-in, a recent version of Eclipse has to be installed. The minimum requirement is version 3.2 of Eclipse. Eclipse comes in several flavors. For new users, we recommend the "classic" version, which includes the Java and plug-in development environment (the latter provides an Error Log view, which can be convenient for diagnosing problems).

Eclipse does not include a Java Runtime Environment (JRE). You will need a 1.4.2 level or higher Java Runtime or Java Development Kit (JDK) installed on your machine in order to run Eclipse. Many systems already include Java. When in doubt, you can always quickly and easily install the latest version through the www.java.com web site.

Eclipse can be downloaded from www.eclipse.org.

4.1.2 Firewall Settings

One of the biggest hindrances for the use of GridFTP are (too) restrictive firewall settings that prevent the required communication between the GridFTP client and servers. In order for the software to work properly, consult your network administrator to make sure that the firewall allows the following types of connections:

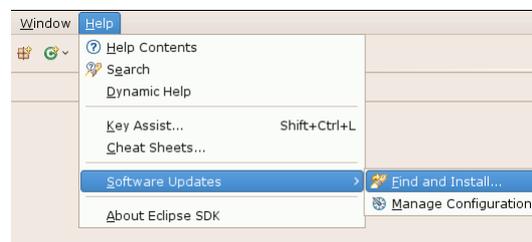
- Every computer on which the GridFTP Client is executed must be able to connect to the TCP port 2811 on the GridFTP servers that you wish to access. This port is used for the "control connection" (sending commands to the server and receiving command acknowledgements).
- Every computer on which the GridFTP Client is executed must be able to connect to a certain range of TCP ports on the GridFTP servers. This range is designated for "data connections" (used for transferring files, but also for reporting directory listings to the client). The exact range is likely prescribed by the Grid infrastructure project to which the servers belong. For example, in

the German Grid the port range 20000-25000 must be reachable from the client.

4.2 Installing the GridFTP Client

The installation is properly performed by using the Eclipse Update Manager. The Update Manager connects to the GridFTP Client Update Site [<http://bi.offis.de/gridftp/updates>]. The Update Site can also be stored locally if no Internet connection is available and can be downloaded from the [GridFTP Client project page](#).

After starting Eclipse, select **Help > Software and Updates > Find and Install...** from the menu.



Find and Install...

Select **Search for new features to install** and press **Next**. Press the **Add Update Site...** button for a remote installation via HTTP or the **Add Local Site...** button if the Update Site is available locally. In a remote installation (recommended), enter any desired **Name** and the **URL** <http://bi.offis.de/gridftp/updates>.



Adding a new update site

Tick the newly created site and press the **Next** button. Follow the instructions of the install wizard. The warning about the installation of an unsigned feature can be ignored. Eclipse has to be restarted after the installation.

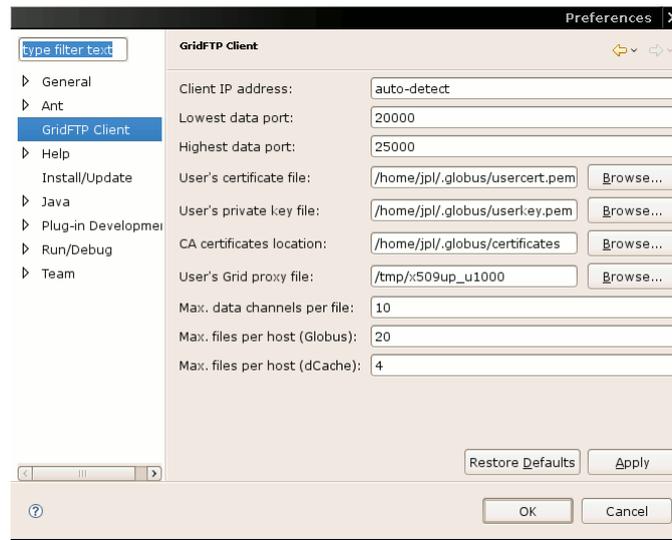
Now the GridFTP Client installation should be complete.

4.3 Uninstalling the GridFTP Client

To temporarily disable or uninstall the currently active version of the GridFTP Client, select **Help > Software and Updates > Manage Configuration** from the menu. Expand the tree in the dialog window which appears and select the GridFTP Client feature. Click on the option **Disable** in the panel on the right side and restart Eclipse when asked. At this point the feature is disabled, but still present on disk. To remove it completely, enter the same dialog again and select the previously disabled version of the GridFTP Client (you may need to toggle showing disabled features in the dialog's toolbar). The option **Uninstall** will now remove the chosen version completely.

5 Initial Configuration

Before setting out to transfer files, to avoid configuration problems, it may be a good idea to have a look at the GridFTP Client's configuration preferences. They can be accessed by selecting **Window > Preferences...** from the Eclipse menu.



Preferences dialog

The meanings of the preferences are quite technical. Fortunately, you can forget about them once they are correctly set, and the automatically guessed values should work without changing on many systems. This is especially true for machines that have been already configured to access Grid resources in the past.

- Client IP address - the IP address of your computer, which is transmitted in communication with GridFTP servers. This is particularly relevant if you wish to use the "active" connection mode where the servers make a connection instead of the other way around. You should leave the setting at its default value of auto-detect unless you experience connectivity problems, in which case you should enter the public (externally accessible) IP address of your machine.
- Lowest data port and highest data port - bounds for the TCP port range used for file transfers and directory listings. This must match whatever port range is unblocked in your local firewall as well as in the accessed GridFTP servers' firewalls. This setting is used to select source ports on the client side. When in doubt, consult your administrator.
- User's certificate file - path to the PEM encoded user certificate, the same certificate file which can be also used by the command-line tool grid-proxy-init.

The default value is guessed depending on your operating system and generally shouldn't need adjusting. This path is only relevant if you wish to use the GridFTP Client to generate the proxy certificate (which is recommended, see below).

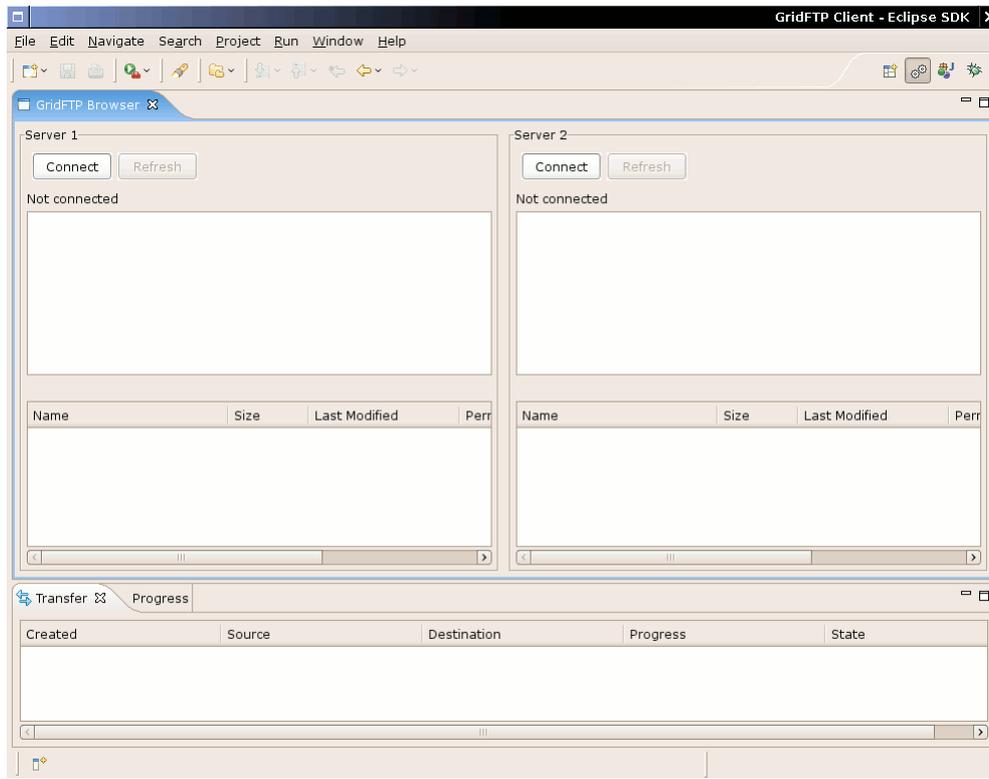
- User's private key file - path to the PEM encoded and encrypted private key, the same key file which can be also used by grid-proxy-init. This goes along with the user certificate, and likewise no changes should be needed.
- CA certificates location - path to a directory which contains the certificates of Certification Authorities used to verify certificates of the GridFTP servers you wish to access. This path is also automatically guessed. When wrong, you will get error messages saying "Unknown CA" on connection attempts. Consult your administrator or the operators of the Grid infrastructure to find out where you can download the required CA certificate files.
- User's Grid proxy file - unlike the above paths, which are for read-only files, this one is read/write: the location of the proxy certificate, which can, but does not have to, be generated by the GridFTP Client from your user certificate, private key and password.
- Max. data channels per file - the maximum number of data channels that should be used within a single file transfer. This setting (like the option -p of globus-url-copy) affects the transfer rate, especially for large files. The GridFTP Client determines the actual number of data channels to use based on the file's size, up to and including the entered setting, with fewer data channels used for small files. Based on our experience, settings as high as 32 can still improve transfer rates. However, your target server might not be configured to accept that many data channels per transfer, in which case transfers would fail with an appropriate error message. For this reason, the default number is 10, which seems to be acceptable out-of-the-box for both Globus and dCache servers.
- Max. files per host - the maximum number of files that may be transferred at the same time from or to a GridFTP host. This can be set for Globus and dCache server types separately. The parameter is intended to keep the servers from being overloaded with requests and reporting connection errors. This has been in particular found out to be a problem for dCache servers. You may try raising the values up to 20 (which is the GridFTP Client's global limit on parallel file transfers) to see whether it improves or hurts the performance in your particular setup.

6 Working with the GridFTP Client

This chapter describes the GridFTP Client's user interface in terms of the supported user activities and explains the concepts underlying the software's functionality.

6.1 The GridFTP Client Perspective

All functionality is most conveniently accessed from a custom Eclipse perspective implemented by the GridFTP Client. You can open this perspective by selecting **Window > Open Perspective > Other...** and selecting the item **GridFTP Client** in the following dialog. When the perspective is active, the screen is by default divided into two areas: the upper area contains a view titled **GridFTP Browser**, while the lower area contains two views, titled **Transfer** and **Progress**:



The **GridFTP Client** perspective

TIP: Should you inadvertently change the layout or content of the GridFTP Client

perspective, the easiest way to restore defaults is to select Window Reset Perspective from the Eclipse menu.

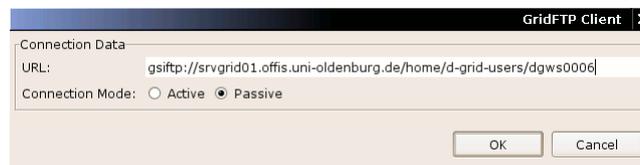
The **GridFTP Browser** view is where you will see the contents of a GridFTP server after connecting. The view is split vertically and contains two functionally equivalent panes (labeled **Server 1** and **Server 2**), so that you can connect to two servers at once to arrange file transfers between them. It does not matter which server is in which pane or in which order you connect. It is also possible to connect to the same server in both panes.

The **Transfer** view is where you will see all currently active (or paused) transfers. The purpose of this view is to provide monitoring information (about the progress and encountered errors) and to allow user-level control of long-running transfers. Concepts associated with transfers are explained in more detail in .

Finally, the **Progress** view is the generic view familiar from other Eclipse perspectives, used to monitor the progress of long-running operations (such as recursive file deletions or disk usage calculations). Active transfers also appear in this view; however, it provides less detailed information and less sophisticated control than the more specialized **Transfer** view.

6.2 Connecting to a GridFTP Server

Click on one of the **Connect** buttons in the **GridFTP Browser** view to open the connection dialog:



The connection dialog

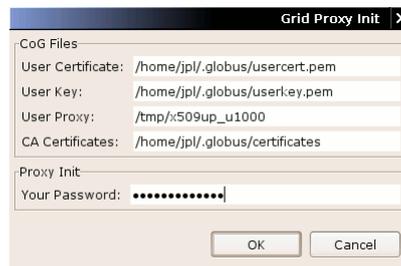
In the **URL** text field, enter a gsiftp:// URL with the hostname of the server, an optional port number, and the initial directory (only subdirectories deeper in the directory tree, but not parent directories will be accessible after connecting). An example of a valid URL would be gsiftp://srvgrid01.offis.uni-oldenburg.de:2811/data/nwp. The gsiftp:// prefix is optional; if not entered, it will be inserted automatically.

The **URL** text field remembers servers to which successful connections were made previously and is filled with the last used server address by default. To make use of this history feature (like in a web browser), simply delete the entire contents of the field by pressing backspace. A list with past server addresses will pop up.

The **Connection Mode** setting controls in which direction data connections (used to obtain directory listings) are established between the GridFTP Client and the

server. The **Passive** mode means that the client will connect to the server. The **Active** mode means that the server will connect to the client. Depending on your firewall settings, one or both modes will work, with the passive mode more likely to work (and thus default). However, we have observed that the passive mode often results in connection errors in combination with the dCache GridFTP server.

When you click **OK**, the GridFTP Client first checks whether a proxy certificate file needed to authenticate on the server already exists (the location of this file can be set in preferences). If the file is not found, a message will be shown and next the authentication dialog will appear:



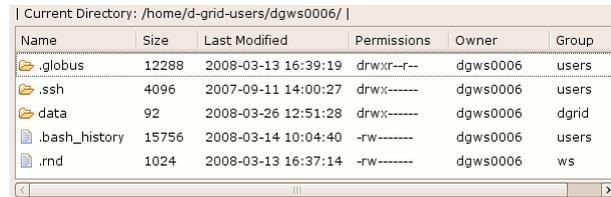
The authentication dialog

You should enter the password which protects your private key (the "grid-proxy-init" password) in this dialog. The other fields are mostly for information, but you might wish to change them to assume a different (non-default) identity for this particular connection.

After the proxy certificate is successfully created (or found), the connection process continues. The chosen GridFTP server is contacted and authentication performed, after which the **GridFTP Browser** view becomes populated with the contents of the initial directory from the previously entered URL. The **Connect** button's label changes to **Disconnect**, with the obvious purpose.

6.3 Browsing a GridFTP Server

After a successful connection, the pane of the **GridFTP Browser** view in which you clicked the **Connect** button first displays the initial directory's contents. You can click on entries in the upper part to expand them and to navigate to a desired directory. Whenever you select some directory in the upper part, the lower part is synchronized to display the files and subdirectories contained in that directory. An alternative method of navigation is to double-click a subdirectory entry in the lower part.



The screenshot shows a window titled "Current Directory: /home/d-grid-users/dgws0006/ |". It contains a table with the following data:

Name	Size	Last Modified	Permissions	Owner	Group
.globus	12288	2008-03-13 16:39:19	drwxr--r--	dgws0006	users
.ssh	4096	2007-09-11 14:00:27	drwx-----	dgws0006	users
data	92	2008-03-26 12:51:28	drwx-----	dgws0006	dgrid
.bash_history	15756	2008-03-14 10:04:40	-rw-----	dgws0006	users
.rmd	1024	2008-03-13 16:37:14	-rw-----	dgws0006	ws

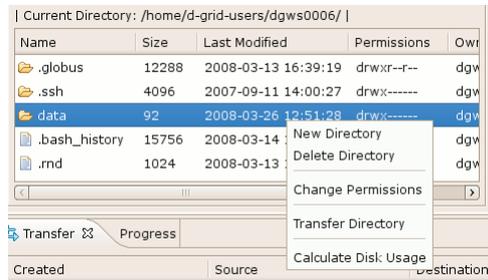
The lower part of the GridFTP Browser

The meanings of the columns in the lower part are as follows:

- Name - name of a file or directory
- Size - size of a file or directory entry in bytes (for directories, this is not the size of contained files or subdirectories; however, the Calculate Disk Usage action provides that size, see below)
- Last Modified - timestamp of the last modification
- Permissions - Unix-style access permissions, formatted as in the output of the Unix command `ls -l`
- Owner - Unix username of the file or directory owner
- Group - Unix group of the file or directory owner

NOTE: The GridFTP implementation of dCache 1.7.0 has the following known limitations which affect the displayed data: Last modified timestamps are neither recorded nor displayed Permissions are shown for the owner only, but not for 'group' or 'world' Neither owner nor group are displayed; however, the invisible owner does in fact affect file permissions

Unlike the upper part, the lower part of the **GridFTP Browser** view supports multiple selections (hold CTRL or SHIFT while clicking on entries). Moreover, it contains a context menu (right-click) which lists several actions:



GridFTP Browser's context menu

The context menu's actions are explained below:

- **New Directory** - creates a new directory in the currently displayed one (selection in the lower part doesn't matter). You will be prompted for the name of the new directory.
- **Delete** - deletes the selected items. If any directories are selected, their contents will be deleted recursively (equivalent to the Unix command `rm -rf`). This is a long-running operation, which is tracked by the **Progress** view. The **GridFTP Browser** view is automatically refreshed after the operation finishes.
- **Change Permissions** - opens a dialog in which you can change the file access permissions for the selected items
- **Transfer** - initiates a transfer that copies the selected items to the directory whose contents are currently displayed in the opposite (left/right) pane of the **GridFTP Browser** view. Selected directories are copied recursively (equivalent to the Unix command `cp -r`). Because copying is in general a long-running operation which may be affected by transient or permanent faults of the servers and client, transfer management is a sophisticated feature in GridFTP Client, which is explained in more detail in .
- **Calculate Disk Usage** - recursively checks the amount of disk space occupied by the selected directories (equivalent to the Unix command `du`). Like deletion, this is a long-running operation. The calculated disk space appears in the upper part of the **GridFTP Browser** view next to the directory names. Some subsequent operations may invalidate the calculation and cause the displayed sizes to disappear.

The **GridFTP Browser** view caches displayed contents to improve responsiveness of the user interface. The **Refresh** button in the top part of the view can be used to discard all cached information displayed in the respective pane and to force obtaining fresh directory listings. This is necessary to see how the GridFTP Server contents were updated by some external application (e.g., another user) and can be also employed to get an updated view of the target server during an active file transfer.

6.4 Transferring Files and Directories

The GridFTP Client is designed to keep track of and to control large transfers consisting of many potentially large-sized files and directories. Such transfers may span hours or even days, and there is significant risk that certain types of failures will occur during the transfer, which need to be addressed (if possible, automatically) in order to ensure the eventual successful completion of the transfer. While the GridFTP Client must be executing in order to keep a transfer alive, it is possible to perform a large transfer in multiple sessions (with breaks in between). When servers go away or software crashes (as it inevitably does), the GridFTP Client does its best to protect your transfers from being aborted in a half-completed, difficult to understand state.

For all of the above reasons transfers, big or small, are a first-class concept in the GridFTP Client - you can create them, see them running, and interact with them during their entire lifetime. A transfer is defined as an operation which copies a set of files and/or subdirectory trees from some source directory on one GridFTP server to some target directory on another GridFTP server. Unlike Unix copy, which overwrites or skips pre-existing target files, the GridFTP Client's transfer works in "append" mode. That means that any existing files in the target location(s) will be treated as incomplete (cut off) versions of their corresponding source files. Only the still missing pieces of the files will be actually transferred, which improves performance when errors are encountered (especially for large files) and makes retrying transfers easy (even after they are completely aborted). It is your responsibility to ensure that the target location is "clear" from files which have a different identity, but the same name as the source files. Note that concurrent overlapping transfers into the same target location are not supported and will likely lead to errors or data corruption. It is your responsibility that your own and someone else's transfers don't collide.

When you start a transfer using the **GridFTP Browser's** context menu, an entry for it appears in the **Transfer** view:



Created	Source	Destination	Progress	State
Wed Mar 26 21:28:04 CET 200	gsiftp://srvgrid01.offis.uni-olde	gsiftp://juggle-glob.fz-juelich.d	18.69 GiB prepar	PREPARING

The **Transfer** view

6.4.1 Transfer States

A transfer is performed in two distinct, consecutive stages: the preparation stage, which gathers information about source files to be copied and creates all target directories, and the copy stage, which actually copies files from their source directories to the appropriate target directories created during the previous stage.

The transfer's stage can be usually deduced from its **State** shown in the **Transfer** view:

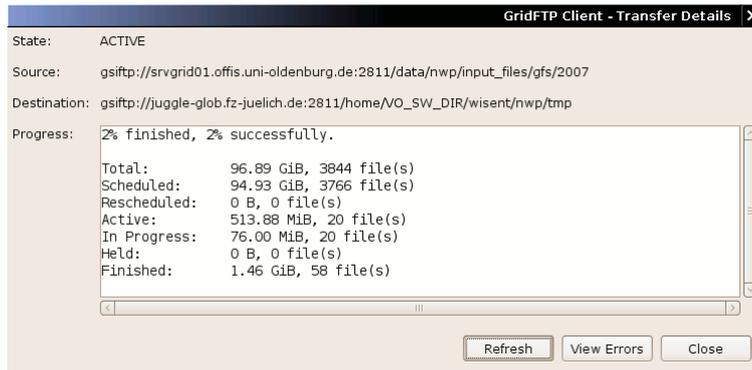
- State **PREPARING** means that the preparation stage is currently underway. The **Progress** column of the **Transfer** view will reflect the total size of source files that have been found so far.
- State **ACTIVE** means that the preparation stage has been successfully completed and the copy stage is underway. The **Progress** column of the **Transfer** view will reflect the completion percent of the transfer. To account for failures, two percentual progress meters are displayed: the first number reflects the proportion of source files that were either successfully transferred or were classified as having encountered non-transient failures. The second number reflects only the proportion of source files that were successfully transferred. In other words, the first number tracks the progress of the transfer's execution - when it reaches 100%, it means that no more files are being transferred, while the second number tracks the progress of the transfer's goal - when it reaches 100%, there are no more files to transfer.

A transfer can also enter three other states:

- State **FINISHED** is equivalent to the second progress meter reaching 100%. Once the transfer reaches this state, there is little more to do other than remove it from the **Transfer** view.
- State **HELD** is equivalent to the first progress meter reaching 100%, with the second progress meter finishing below 100%. In other words, some of the files have not been successfully transferred (yet), and the transfer is suspended. There are only two scenarios that can lead to this situation:
 - You have explicitly paused the transfer.
 - All transferable files have been either successfully transferred or encountered a non-transient failure. Non-transient failures are such that could not be simply worked around by retrying the file's transfer; they require your intervention. For example, a missing target directory is a non-transient failure (you have to check manually why the expected target directory is not there and create it), while temporary network connectivity problems are not. The GridFTP Client contains some built-in expert rules used to classify failures in this manner. In the current version, unrecognized failures are generally classified as non-transient to provide early warnings to the user. In the future, depending on user experiences, this policy may be reversed.
- State **WILL HOLD** occurs when the first progress meter exceeds the second progress meter. In other words, some files have encountered non-transient failures, but there are still other files that can be transferred. This state is meant as an early indicator to make the user aware that a manual intervention will be eventually needed to repair the transfer.

6.4.2 Transfer Details

You can obtain more extensive information about a transfer, which is particularly useful if the transfer unexpectedly reaches the **WILL HOLD** or **HELD** state. Double-click the transfer's entry in the **Transfer** view to open the transfer details dialog:



The transfer details dialog

Apart from the obvious information, the dialog contains a **Progress** text area which reports the following statistics:

- Both progress meters, as explained above.
- Total - the total size and number of all files within the transfer
- Scheduled - the total size and number of files that are not being actively transferred yet, but are awaiting their turn in the queue.
- Rescheduled - the total size and number of files that have reentered the waiting queue (because of errors or user actions).
- Active - the total size and number of files that are being actively transferred. The GridFTP Client transfers up to 20 files concurrently.
- In Progress - the size of the already transferred part of the active files.
- Held - the total size and number of files that encountered non-transient failures or were held when the transfer was suspended on user's request.
- Finished - the total size and number of successfully transferred files.

You can click on the **Refresh** button to update the displayed statistics. The button **View Errors** opens a new dialog, which allows you to browse the errors encountered by the transfer as a whole (i.e., during the preparation stage) or during attempts to transfer the individual source files. The current version of the GridFTP Client simply displays unadorned exception stack traces. These messages are certainly easier to understand to administrators and software

developers than to end users. However, by inspecting them, you might find hints about the actual cause of the problem (e.g., "Permission denied" somewhere in the message would suggest that you should examine the source file's and target directory's permissions and ownership). Future versions will hopefully improve the quality of the reported error messages, especially as we accumulate insights into possible failure scenarios.

6.4.3 Controlling Transfers

The **Transfer** view contains a context menu (right-click) which allows you one of the actions that affect the selected transfer's state:

- **Hold** - (only enabled for active transfers) temporarily suspends the transfer and causes it to enter the **HELD** state. This action is also executed if you "cancel" the transfer operation displayed in the **Progress** view. Note that if you hold a transfer in the **PREPARING** state, it will have to repeat the entire preparation stage, which may take a while depending on the number of source files and directories.
- **Resume** - reschedules all files that were held: another attempt will be made to copy these files. This effectively resumes all (non-finished) files from the transfer, regardless of whether they got hold on your request or because of some encountered failures. In the latter case, the failures might reoccur, and the transfer might become held again after a while.
- **Remove** - removes the transfer from the **Transfer** view. This is the last action in every transfer's lifetime and not reversible. If the transfer has not finished yet, it will be interrupted and removed. Note that the files that have been already transferred will not be automatically deleted in any case - it is up to you to clean up if necessary.

6.4.4 Transfer Persistence

All transfers recorded in the **Transfer** view survive the termination of the GridFTP Client - regardless of whether you normally close the application or whether it crashes for some reason. After you restart Eclipse, the GridFTP Client will attempt to reconnect to the GridFTP servers of any recorded transfers and after a short while the **Transfer** view will become repopulated. Transfers that were active beforehand will automatically start running (you may need to minimize their progress dialogs), other transfers will remain in their respective states (**HELD** or **FINISHED**).

7 Obtaining Help

The [Bugs](#) and [Feature Requests](#) trackers on SourceForge contain descriptions of the current open issues for the GridFTP Client. If you encounter a problem, you can also search the [Help forum](#) for previous reports and solutions and ask your question there.

8 Acknowledgements

The GridFTP Client was originally developed at the [OFFIS](#) Institute for Information Technology within the research project [WISENT](#), sponsored by the German Federal Ministry of Education and Research (BMBF) as part of the D-Grid initiative under grant No. 01C5968.

We would like to thank the [Jülich Supercomputing Centre](#) for supporting our tests with different versions and configurations of dCache.

The GridFTP Client relies heavily on [CoG JGlobus](#) library provided by the Globus project. Thanks to the JGlobus team for providing such a stable foundation.