Introduction and Description

The GTP (GRAVITE Transfer Protocol) is a system designed to provide a single, consistent interface for data discovery and retrieval among one or more disparate information storage systems.

The GTP client utility has two main modes of operation. One mode retrieves a list of file records from the server database. The other mode provides a means of downloading the physical files corresponding to those database records.

The client tool is comprised of two main components: the GTPProxy program and the GTPClient program. Both of these are executable Java JAR files. The purpose of GTPProxy is to provide a persistent communications gateway to the data server in the background while enabling the user to invoke queries to the server while still retaining access to the shell prompt. This model allows for customized scripting and incorporation with other programming languages.

A Linux shell script named "`gtp`" is provided to simplify the usage of these two components and acts as the point of execution for all client commands.

System Requirements

A Java 6 or newer runtime environment is required to run the compiled bytecode as the client tool and its contents are compiled against Java 1.6 build 11. A Linux shell script is provided to simplify usage and squelch any unnecessary or extraneous log messages at runtime. The Java binary (`java`) should be in the environment path.

Port 1099 is required locally by the client components so that they may communicate. Communication between the two modules is performed via Java RMI (Remote Method Invocation).

Download and Installation

The latest GTP client (packaged as a tar file) may be downloaded from CasaNOSA Subversion repository within the "GRAVITE Data Portal" project:
https://casanosa.noaa.gov/svn/gtp/old/trunk/dist/

If you need help with your CasaNOSA account, please contact the helpdesk at 301-713-9300. The Helpdesk can also you provide you with your GTP username and password. To change your GTP password, navigate to:
https://gravite.ipo.noaa.gov/gtp-pw/changepass

Once the client is downloaded, untar the GTP client and setup environment variables:

```
mkdir /home/username/gtpclient
cp GTPClient.tar.gz /home/username/gtpclient
cd /home/username/gtpclient
tar zxvf GTPClient.tar.gz
```

Add these lines to your `/home/username/.cshrc` file:

`setenv JAVA_HOME /opt/jdk1.6.0_14` (should point to your installed Java SDK)

```
setenv GTP_HOME /home/username/gtpclient
alias gtp `/home/username/gtpclient/gtp'
```

Next, execute "`source ~/.cshrc`".

If you wish, you can edit the "gtpclient/gtp.properties file to include your username and/or password. For example,

gtp.username=jsmith
gtp.password=abc123

Document Conventions

Various conventions used in this document:

`command subcommand arg1 arg2 argN`

All commands, subcommands, keyword arguments and output are represented in this way.

`< parameter1 | parameter2 >`

Required arguments are surrounded by angled brackets.

`parameter`

Any variable argument will be represented by italicized typeface.

`[ parameter1 | parameter2 ]`

Parameters appearing between square brackets denote optional arguments. Option sets may include variable parameters.

Items separated by the pipe character cannot be used in conjunction with each other. They represent an available option for one specific argument.

Usage

Overview:

There are four primary subcommands that determine the mode of the tool. They are `auth`, `exit`, `list` and `download`.

Before the `list` and `download` commands can be of any use, the user must set up a session with the server by invoking `gtp auth`. The two main modes of operation include `list`, which allows the user to view the contents available on the server and `download`, which allows the user to retrieve said contents. Once the user/script is finished with these modes, it is advisable to perform `gtp exit` in order to stop the GTPProxy daemon and end the session with the server. Session life is determined by the server. Leaving the GTPProxy running does not guarantee availability of the server.

Authentication (`gtp auth`):

In order to use the tool, the user must set up a session with the server. All communication between the client and server is encrypted over HTTPS on port 443.

```
gtp auth
```

The user will be prompted for a username and password--and any other credentials depending on the server being accessed.
Upon successful authentication, the session will be available to perform the `list` and `download` subcommands.

Record Discovery (`gtp list`):

Retrieve list of available files on server.

```
gtp list < newest
         | oldest
         | since <id record_id | ts timestamp >
         | between < id start_id end_id | ts start_ts end_ts >
         >
```

Argument	Description/Arguments
`newest`	Retrieve the newest record in the database
`oldest`	Retrieve the oldest record in the database
`since`	Retrieve records in ascending chronological order starting from

<table>
<tr><td></td><td>either record_id or timestamp.
<table>
<tr><th>Argument</th><th>Type</th><th>Description</th></tr>
<tr><td>record_id</td><td>64-bit unsigned integer</td><td>ID corresponding to a file’s database record</td></tr>
<tr><td>timestamp</td><td>32-bit unsigned integer</td><td>No. of seconds since Unix epoch. (Jan 1st 1970 00:00:00 UTC)</td></tr>
</table></td></tr>
<tr><td>between</td><td>Retrieve records between having an id or timestamp between start_* and end_*
<table>
<tr><th>Argument</th><th>Type</th><th>Description</th></tr>
<tr><td>start_id</td><td>64-bit unsigned integer</td><td>ID corresponding to a file’s database record. Min value in range.</td></tr>
<tr><td>end_id</td><td>64-bit unsigned integer</td><td>ID corresponding to a file’s database record. Max value in range.</td></tr>
<tr><td>start_ts</td><td>32-bit unsigned integer</td><td>No. of seconds since Unix epoch. (Jan 1st 1970 00:00:00 UTC). Min value in range.</td></tr>
<tr><td>end_ts</td><td>32-bit unsigned integer</td><td>No. of seconds since Unix epoch. (Jan 1st 1970 00:00:00 UTC). Max value in range.</td></tr>
</table></td></tr>
</table>

`gtp list` queries may be filtered by data type. Available data types may be listed using `gtp keyorder`:

```
gtp keyorder type
gtp ls
```

To query by file type:

```
gtp list type=<data type>
```

Downloading Files (`gtp download`):

Download files from server to local system.

`gtp download <file | other_file_type> <record_id> [target]`

Argument	Type	Description
`other_file_type`	string*	Download files associated with source file with id {id}. (e.g. RIP DDR)
`record_id`	64-bit unsigned integer	ID corresponding to a file's database record
`target`	string*	Optional local file path and name. (e.g. /home/commodore/my_rip.file). The target path defaults to the current working directory. Also, by default, all files retain their original names unless otherwise specified. (See Examples below.)

* Character limit is the lesser of ((2^31)-1) or (bytes allotted to Java memory heap)/2 bytes. (Java uses 16-bit Unicode character representation.)

Closing Session (`gtp exit`):

End GTPProxy and close session.

`gtp exit`

Output Format

GTPClient streams all output to standard output and standard error.

`record_id filename checksum size`

Output delimiters are as follows:
Word separator: space character (ASCII 32)
Line separator: System-dependent. From the Java API: "The line separator string is defined by the system property `line.separator`, and is not necessarily a single newline character." Typically, the line separator is '\n' on Linux, '\r\n' on Windows, etc.

Error Codes

All error messages are directed to the system's standard error stream along with error-specific return codes. Successful operations return code 0, while all other errors return positive integers corresponding to the following table:

Return (Exit)	Explanation

Code	
0	Successful operation. No errors encountered.
1	Could not connect to server. Timed out.
2	Connected to server. Authentication failed.
3	Unexpected result from server on web service invocation.
4	Remote database error.
5	GTPProxy daemon is not running.
6	Error downloading file: unknown error.
7	Error downloading file: could not create path.
8	Error downloading file: insufficient privileges.
9	Out of memory.
10	Unexpected, unknown error.
11	TBD
12	TBD
13	TBD

The exit code is available in most shell environments via a predefined variable. For instance, in a Bash environment, the exit code for the last executed command can be

retrieved via the variable '`$?`'. Scripting languages like PHP and Perl offer language-specific methods for retrieving this value from forked, system-level executions.

Examples

To get a list of files that were loaded into the database AFTER file ID was loaded 398388 one would invoke the following:

```
gtp list since id 398388
```

Result:

```
398393 IVAOT_npp_d20030125_..._SCI_194.h5 8c51e25b94c49c1b1647b753c694f431 164777904
398391 IVAOT_npp_d20030125_..._SCI_194.h5 7983f755c9ff9094190546d73ecf6c19 164777600
398389 IVAOT_npp_d20030125_..._SCI_194.h5 4decf51d05b458161dac9f8b14fe44e5 164777392
```

To get a list of files loaded into the database AFTER the unix timestamp 1232301674 (01/18/09 @ 12:01pm):

```
gtp list since ts 1232301674
```

Result:

```
398393 IVAOT_npp_d20030125_..._SCI_194.h5 8c51e25b94c49c1b1647b753c694f431 164777904
398391 IVAOT_npp_d20030125_..._SCI_194.h5 7983f755c9ff9094190546d73ecf6c19 164777600
398389 IVAOT_npp_d20030125_..._SCI_194.h5 4decf51d05b458161dac9f8b14fe44e5 164777392
```

To get the oldest available file record:

```
gtp list oldest
```

Result:

```
15755 VMUGE_npp_d20030125_..._SCI_10.h5 122b8a0cefa8b9876455af2ca57a836b 127401344
```

To get the newest available file record:

```
gtp list newest
```

Result:

```
398393 IVAOT_npp_d20030125_..._SCI_194.h5 8c51e25b94c49c1b1647b753c694f431 164777904
```

To download a file corresponding to record id 398389 to the current directory:

```
gtp download file 398389
```

To download a file corresponding to record id 398391 to a new subdirectory called 'new_files':

```
gtp download file 398391 new_files/
```

Note: The trailing slash is necessary to create the directory. Otherwise, the file will end up being called 'new_files.'

To download a file corresponding to record id 398391 as 'a_file.h5' to a new subdirectory called 'files':

```
gtp download file 398391 files/a_file.h5
```

To list the newest 10 files of type “IICMO_NPP”:

```
gtp list newest 10 type=IICMO_NPP
```