GTP User's Guide
Last Revised October 19, 2011

Introduction and Description

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

Specifically, the GTP allows users to query and download data files from the GRAVITE Investigator Computing Facility (ICF), such as proxy and test data (pre-launch), NPP mission data, and other data to support calibration and validation (cal/val) post-launch.

The GTP client utility has two main modes of operation. One mode retrieves a list of file records from the GRAVITE ICF server database, referred to as "DSD," the GRAVITE Data Storage and Distribution system. The other mode provides a means of downloading the physical files corresponding to those database records. Each file available for query and download has its own associated record in the DSD database. Each record also has associated metadata that may be used to query for specific files on interest.

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).

GTP communicates with GRAVITE and pulls data over HTTPS. To test your connection to the GRAVITE system, use a web browser to navigate to https://gravite.ipo.noaa.gov. If the connection is established, you will see a NOAA logo.

Getting Help and Account Administration

New account requests should be sent to Janet.T.Fennema@nasa.gov by the team lead.

For GTP password issues, contact the Helpdesk by phone at 301-713-9300, between the hours of 9am and 5pm Eastern Time, Monday to Friday. When communicating with the Helpdesk, please be sure to reference "GTP" as the Helpdesk also manages CasaNOSA accounts, which are a separate set of login credentials.

GTP passwords may be changed via the website below.
https://archivist.ipo.noaa.gov/ArchivistProject/

Note that the authentication mechanisms for the GTP and Cal/Val Findings Tool are combined; so a user with both accounts can use the same username and password to login to both systems. If a password is changed, the new password must be used to access both the GTP and Cal/Val Findings Tool.

For GTP questions, problems, or comments, contact Richard.Yu@nasa.gov.

Download and Installation

The latest GTP client, packaged as a tar file, may be downloaded from the CasaNOSA Subversion repository within the "GRAVITE Data Portal" project. Use your CasaNOSA account username and password to access the link below (Contact 301-713-9300 for CasaNOSA help). As of October 19, 2011, the current version 1.0.0. GTPClient-1.0.0.tar.gz is the only file you need to download.

https://casanosa.noaa.gov/svn/gtp/old/trunk/dist/

If using a Linux operating system:

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

```
mkdir /home/username/gtpclient
cp GTPClient-1.0.0.tar.gz /home/username/gtpclient
cd /home/username/gtpclient
tar zxvf GTPClient-1.0.0.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 using a Windows operating system:

To untar the client, you can use a freeware tool called PeaZip. You will need to set two Windows environment variables, GTP_HOME and JAVA_HOME, the first pointing to the location of the GTP client (i.e. C:\gtpclient) and the second to your Java package (i.e. C:\Program Files (x86)\Java\jre1.6.0_22).

To add an environment variable in Windows XP, go to:

Start -> My Computer (right click) -> Properties -> Advanced tab -> Environment Variables -> New (User Variable)

To add an environment variable in Windows 7:

Click the “Start” button and type “variables” in the “Search programs and files” bar. Click “Edit environment variables for your account”. Under user variables for your account, select “New”.

A Cygwin or Putty tool is not required to run to GTP, a Windows command line will work (go to Start -> Run -> type "cmd" and hit Enter). Navigate to the GTP home directory at the command line before attempting to connect.

Note: GTP will run in a Mac environment as well, but detailed instructions are not yet available. Follow similar procedures to untar the package and set environment variables before attempting to connect.

Configuration

GTP may be configured by editing the "gtp.properties" file in your GTP home directory.

Variable	Description
proxytimeout	Defines the GTPProxy timeout in seconds. The default value is 0, which is no timeout.
gtp.logging	The default behavior is no logging. To turn logging on, set gtp.logging=1. Logs will be written to the "log" directory in your GTP home directory.
gtp.username	(Optional) Set this variable to username to avoid being prompted for your username each time you attempt to authenticate.
gtp.password	(Optional) If gtp.username is set, set the gtp.password variable to avoid being prompted for your password each time you attempt to authenticate.
displayMode	There are three options for displaying download information: none, static and full. "none" = no output "static" = displays file id, download speed (KB/s), and time elapsed after the download completes "full" = displays in real time the file id, progress bar, percentage completed, download speed, and time elapsed. The default mode is "full," but it is not ideal for file output. If you are redirecting output to a file, it is suggested that "static" or "none" modes be used.

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.

GTP Basic 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. To authenticate, execute the command:

```
gtp auth
```

The user will be prompted for a username and password unless they are pre-set in the gtp.properties file.

Upon successful authentication, the session will be available to perform the list and download subcommands.

Record Discovery (gtp list):

To retrieve a list of available files on the 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

<table>
<tr><td>since</td><td>Retrieve records in ascending chronological order starting from 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>

Note: The timestamps used in the above queries correspond to the observation time of a data product.

Conditional Record Discovery (Metadata):

Each record has associated metadata that may be used for record discovery in place of or prior to the newest, oldest, since, and between parameters described above. To return a list of metadata fields available for query, execute:

```
gtp keylist
```

For example, each record has an associated "type" metadata field. For a list of types and corresponding descriptions, visit the "Data Types Wiki" on CasaNOSA here:

https://casanosa.noaa.gov/docman/wiki_doc.php?id=2574&group_id=342

Available data types, without descriptions, may also be seen by issuing this series of commands:

```
gtp keyorder type
gtp ls
```

Users may query against a single metadata field using a logical operator. Users may query against multiple metadata fields using conditional operators. These qualifiers must be included in the GTP query as a string of formatted qualifiers:

```
gtp list < qualifiers >
```

For example, to query for all records of type "XYZ":

```
gtp list type=XYZ
```

More examples are provided later.

Downloading Files (gtp download):

To download files from the server to local system:

```
gtp download <record_id> [target]
```

Argument	Type	Description
record_id	64-bit unsigned integer	ID corresponding to a file's database record
target	string*	Optional local file path and/or file 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.

* 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):

To terminate GTPProxy and close the session, execute:

```
gtp exit
```

Output Format:

GTPClient streams all output to standard output and standard error. Each line contains four values:

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) Code	**Explanation**
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.

Additional Commands:

To display the help screen:

```
gtp help
```

To display the current version of the GTP client:

```
gtp version
```

Wildcards:

The percent symbol (%) is the wildcard used by GTP to represent a string of characters or a single character. The wildcard may be used with text-based metadata parameters (for example, the metadata parameter "name" is the filename). See Example 13 below.

NPP data product filenames do include the five-character product shortname(s) in addition to data observation times, file creation time, and orbit number. A user could query against these metrics by using wildcards to search through all filenames on GRAVITE. However, this method is inefficient and such queries may time-out.

Product type, data observation times, file creation times, and orbit numbers are all indexed metadata parameters pre-populated in the GRAVITE database. Users who wish to query against these metrics should build their GTP queries using the conditional record discovery methods outlined in this guide, **not** by using wildcards to query against the filename.

Examples:

1. To obtain a list of files that were ingested into the database AFTER record ID 398388:

```
gtp list since id 398388
```

2. To obtain a list of files with an observation time AFTER the UNIX timestamp 1232301674 (01/18/2009 @ 12:01pm):

```
gtp list since ts 1232301674      OR
gtp list "start_sec>1232301674"
```

3. To obtain a list of files ingested into the database AFTER the UNIX timestamp 1232301674 (01/18/2009 @ 12:01pm):

```
gtp list "received>1232301674"
```

4. To obtain a list of files with an observation time AFTER the UNIX timestamp 1232301674 (01/18/2009 @ 12:01pm) and BEFORE the UNIX timestamp 1232301734 (01/18/2009 @ 12:02pm):

```
gtp list between ts 1232301674 1232301734      OR
gtp list "[start_sec>1232301674&end_sec<1232301734]"
```

5. To obtain a list of files ingested into the database AFTER the UNIX timestamp 1232301674 (01/18/2009 @ 12:01pm) and BEFORE the UNIX timestamp 1232301734 (01/18/2009 @ 12:02pm):

```
gtp list "[received>1232301674&received<1232301734]"
```

6. To obtain a list of the oldest five records in the database:

```
gtp list oldest 5
```

7. To obtain a list of the newest ten records in the database:

```
gtp list newest 10
```

8. To download the file with record id 398389 to the current directory:

```
gtp download file 398389
```

9. To download the file with 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.'

10. To download the file with record id 398391 as “a_file.h5” to a new subdirectory called “files”:

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

11. To obtain a list of the newest ten files of type “SCRIS_NPP” (a CrIS Science SDR):

```
gtp list newest 10 type=SCRIS_NPP
```

12. To obtain a list of files of type VCBHO_NPP with an observation time between UNIX timestamp 1232301674 (01/18/2009 @ 12:01pm) and UNIX timestamp 1232452860 (01/20/2009 @ 12:01pm):

```
gtp list between ts 1232301674 1232452860 type=VCBHO_NPP
```

OR

```
gtp list "type=VCBHO_NPP&[start_sec>1232301674&end_sec<1232452860]"
```

13. To obtain a list of the files in the database that have the string "d20100413_t1215470_e1216580" in the filename:

```
gtp list "name=%d20100413_t1215470_e1216580%"
```

GTP Advanced Usage

This section with discuss complex conditional queries, the virtual directory structure feature, and abilities to customize output and date formats.

If a user is looking for a very specific dataset, one can use the virtual data structure to browse the repository and/or build sophisticated conditional queries for scripting purposes.

Virtual Directory Structure:

To setup and use the virtual directory structure using metadata fields, use these commands:

To return a list of available metadata fields:

```
gtp keylist
```

To set the virtual directory structure:

```
gtp keyorder <comma delimited list of metadata fields>
```

To display the contents of the current virtual directory [Note: Keyorder must be set first]:

```
gtp ls
```

To navigate through a virtual directory [Note: Use ".." to move up a directory]:

```
gtp cd <directory path>
```

To display the current working virtual directory:

```
gtp pwd
```

For example,

To list night granules for files of type SVM03_AQUA, with a start scan between May 10, 2010, 2am GMT, and May 10, 2010, 3am GMT:

```
gtp keyorder type,daynight,start_sec
gtp cd SVM03_AQUA
gtp ls [to view a list of daynight classifications]
gtp cd night
gtp cd 1273456800 to 1273460400
```

`gtp pwd` [to print the current virtual directory path]
`gtp ls` [to list the files]

Complex Conditional Queries:

Example 1: Get a list of night granules for files of type SVM03_AQUA, with a start scan after May 10, 2010, 2am GMT, and an end scan prior to May 10, 2010, 3am GMT:

```
gtp list "type=SVM03_AQUA&daynight=night&[start_sec>1273456800&end_sec<1273460400]"
```

OR

```
gtp list between ts 1273456800 1273460400 "[type=SVM03_AQUA&daynight=night]"
```

Example 2: Get a list of the ten most recent files such that the type is GDNBO_AQUA *or* the filename follows "SV%_npp%.h%" and it is a night granule:

```
gtp list newest 10 "[[[name=SV%_npp%.h%]&daynight=night]|type=GDNBO_AQUA]"
```

Example 3: Get a list of files of type SVM01_AQUA or SVM02_AQUA (VIIRS M1 and M2 Band proxy SDRs), containing data with a northern most latitude of 50 degrees, a southern most latitude of 40 degrees, a western most longitude of 10 degrees, and an eastern most longitude of 5 degrees.

```
gtp list "[northlat<50&southlat>40&westlon<10&eastlon>5&[type=SVM01_AQUA|type=SVM02_AQUA]]"
```

Customize the Standard Output:

Users may replace the standard "*record_id filename checksum size*" output with a custom output defined by a set of metadata fields.

To set the custom output:

```
gtp setoutput "<comma-delimited list of metadata fields>"
```

To revert output to the default (id,name,checksum,size):

```
gtp clearoutput
```

Customize the Date Format

Users may alter the data format to something other than the standard UNIX timestamp.

To customize the date format:

```
gtp dateformat [<valid date format>]
```

Example: `gtp dateformat yyyy-MM-dd HH:mm:ss`

Visit http://java.sun.com/j2se/1.5.0/docs/api/java/text/SimpleDateFormat.html for information on acceptable formats.

If the date format is not provided, gtp will display the current format or indicate none has been specified.

When specifying a date/time in a query with a customized date format, be sure to place the date/time in double quotes so that the query is properly parsed. Dates/times with space characters will not be properly parsed if they contain one or more space characters. See Example 2 below.

To remove the custom date format:

`gtp cleardateformat`

Example 1 -

```
[mdenning@dev10 ~]$ gtp list newest 2 type=GMODO_AQUA
13040996
GMODO_npp_d20100701_t1221233_e1222339_b43401_c20100701133051513450_noaa_ada.h5
ee1906575f4cf19950465c871d284508 78666448
13040995
GMODO_npp_d20100701_t1222342_e1223448_b43401_c20100701133051283621_noaa_ada.h5
069f1e379da8352b73a994191cb91d26 78666448
[mdenning@dev10 ~]$ gtp setoutput "id, start_sec, end_sec, type, daynight"
[mdenning@dev10 ~]$ gtp list newest 2 type=GMODO_AQUA
13040996 1277986891 1277986961 GMODO_AQUA night
13040995 1277986962 1277987031 GMODO_AQUA night
[mdenning@dev10 ~]$ gtp dateformat yyyy-MM-dd HH:mm:ss
[mdenning@dev10 ~]$ gtp list newest 2 type=GMODO_AQUA
13040996 2010-07-01 07:21:31 2010-07-01 07:22:41 GMODO_AQUA night
13040995 2010-07-01 07:22:42 2010-07-01 07:23:51 GMODO_AQUA night
[mdenning@dev10 ~]$ gtp cleardateformat
[mdenning@dev10 ~]$ gtp clearoutput
[mdenning@dev10 ~]$ gtp list newest 2 type=GMODO_AQUA
13040996
GMODO_npp_d20100701_t1221233_e1222339_b43401_c20100701133051513450_noaa_ada.h5
ee1906575f4cf19950465c871d284508 78666448
13040995
GMODO_npp_d20100701_t1222342_e1223448_b43401_c20100701133051283621_noaa_ada.h5
069f1e379da8352b73a994191cb91d26 78666448
```

Example 2 -

```
[mdenning@dev10 ~]$ gtp list between ts 1277877600 1277878200 type=GMODO_AQUA
12956600
GMODO_npp_d20100630_t0231147_e0232253_b43381_c20100630020050835608_noaa_ada.h5
ed78323788682f8e97a6807cbaae23af 78666448
12956596
GMODO_npp_d20100630_t0230038_e0231144_b43380_c20100630020049993135_noaa_ada.h5
650fae253c782c7de2f5ed74c442039b 78666448
[mdenning@dev10 ~]$ gtp dateformat yyyy-MM-dd HH:mm
```

```
[mdenning@dev10 ~]$ gtp list between ts "2010-06-30 01:00" "2010-06-30 01:10"
type=GMODO_AQUA
12956600
GMODO_npp_d20100630_t0231147_e0232253_b43381_c20100630020050835608_noaa_ada.h5
ed78323788682f8e97a6807cbaae23af 78666448
12956596
GMODO_npp_d20100630_t0230038_e0231144_b43380_c20100630020049993135_noaa_ada.h5
650fae253c782c7de2f5ed74c442039b 78666448
```