

Globus Online - Catalog Command Line Client

[<https://github.com/globusonline/catalog-client>]

Catalog User Interface: <https://catalog-alpha.globuscs.info/>

Contact: Ben Blaiszik (blaiszik@uchicago.edu)

OBTAIN GLOBUS CREDENTIALS

- 1) Test Instance: <https://test.globuscs.info/SignUp>
 - 2) Production Instance: <https://www.globus.org/SignUp> - **this is the credential you will use with Catalog**
-

INSTALL REQUIREMENTS

- 1) Install Globus Catalog-client (<https://github.com/globusonline/catalog-client>)

```
git clone https://github.com/globusonline/catalog-client
cd catalog-client
python setup.py install --user
```

- 2) Install Globus Transfer API(<https://github.com/globusonline/transfer-api-client-python>)

```
git clone https://github.com/globusonline/transfer-api-client-python
cd transfer-api-client-python
python setup.py install --user
```

CLI Example Bash Script

This shell script shows several examples of commands that can be used via the CLI

```
cd catalog-client/globusonline/catalog/client/examples
sh test_command_line.sh
```

QUICK START

1. Follow **INSTALL REQUIREMENTS** (top section - above)
2. Navigate to the directory you installed catalog-client and cd globusonline/catalog/client/examples to find the python script catalog.py
3. **Create a catalog**

```
Bens-MacBook-Pro:examples ben$ python catalog.py create_catalog 'My New Catalog'
92
```

- a. Run the command: `python catalog.py create_catalog MyCatalog` or `python catalog.py create_catalog 'My Catalog Name'` if you there are spaces. Upon success, this command will return your catalog ID. Save this ID programmatically, or remember this value.
- b. You can save a default catalog in an environment variable. With a default catalog, you do not need to always enter which catalog the commands should be applied to. To set a

default catalog, set the environment variable GCAT_DEFAULT_CATALOG. From the command line: `export GCAT_DEFAULT_CATALOG_ID=17`

```
Bens-MacBook-Pro:examples ben$ export GCAT_DEFAULT_CATALOG_ID=17
Bens-MacBook-Pro:examples ben$ echo $GCAT_DEFAULT_CATALOG_ID
17
```

4. **Authenticate:** If this is your first time running a command through the CLI, the catalog client will ask you to authenticate before executing. Authenticate with Globus test credentials. If you do not have Globus test credentials, create them here <https://test.globuscs.info/>

5. **Add a dataset** to your new catalog

```
Bens-MacBook-Pro:examples ben$ python catalog.py create_dataset 'My New Dataset'
52
```

- a. Run the command: `python catalog.py create_dataset 'My New Dataset'`
- b. Note that 52 is the Dataset ID of your newly created catalog
- c. Upon success, this command will return the new Dataset ID. Save this ID programmatically to refer directly to this dataset.

6. **Add an annotation** to your dataset

- a. If your annotation is text
 - i. Add by dataset name: `python catalog.py add_dataset_annotation -n 'My New Dataset' new_annotation:the_value`
** The dataset name must resolve uniquely within the catalog*
 - ii. Add by dataset ID: `python catalog.py add_dataset_annotation 52 new_annotation:the_value`

```
Bens-MacBook-Pro:examples ben$ python catalog.py add_dataset_annotation 52 new_annotation:the_value
52
{'message': u'Annotations added successfully', u'code': u'Added', u'request_id': u'XLLVPFWVr'}
```

- b. If your annotation is typed as other than text (i.e. an integer, boolean, or date)
 - i. Create the annotation definition. This annotation definition will then act as a shared vocabulary that can be used by all elements within your new catalog
 1. Run the command: `python catalog.py create_annotation_def 66 'myAnnotation' 'text'`
 2. See the list of valid data types below
 3. Apply your annotation as in step a above

```
mcswl188:examples argonne$ python catalog.py create_annotation_def 66 'myAnnotation' 'text'
CREATE ANNOTATION DEF - Catalog ID:66 Name:myAnnotation Type:text
RestResult(response=<httplib.HTTPResponse instance at 0x10191e908>, body={u'writepolicy': u'anonymou', u'name': u'myAnnotation', u'read users': u'*', u'readpolicy': u'anonymou', u'value_type': u'text', u'multivalued': False, u'unique': False})
```

Alternate input format: you may now also use the form:

```
python catalog.py add_dataset_annotation -t 2506 PI:"Ray Osborn" X:3 Y:
```

This would create three annotations. The quotes are only required for shell parsing of the space in the name. `Y:` results in the empty string as the value.

The default (JSON) format is triggered if the argument starts with '{'

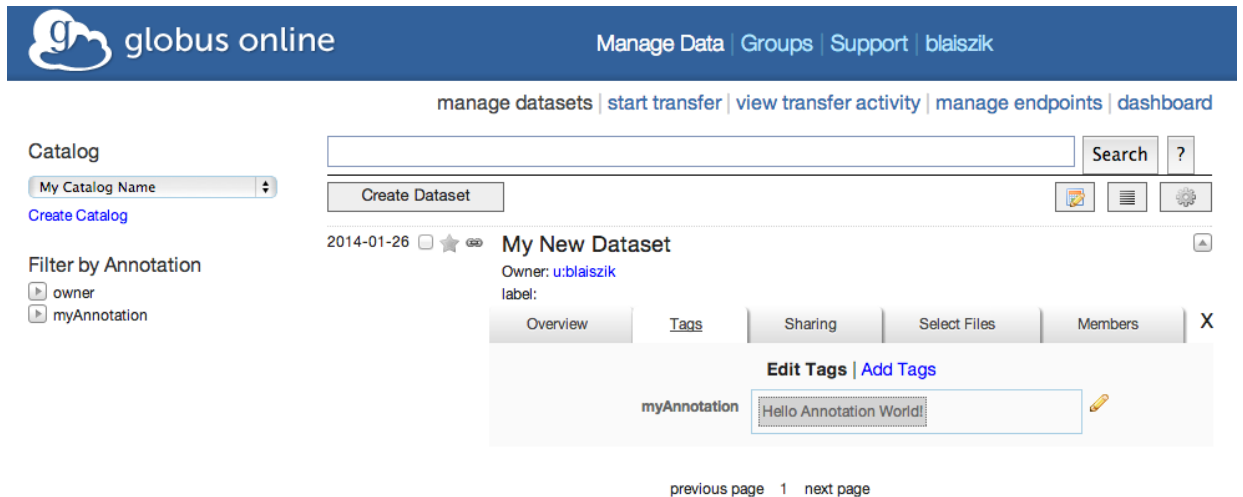
7. Find your dataset based on an annotation

- a. Run the command `python catalog.py query_datasets myAnnotation LIKE %`
 - i. This will return all datasets in your catalog that contain ANY myAnnotation annotation.
 - ii. See documentation below for other query capabilities

```
mcswl188:examples argonne$ python catalog.py query_datasets 66 myAnnotation LIKE %  
[{"name": "My New Dataset", "created": "2014-01-27 16:11:24.314194+00", "favorite": null, "readok": true, "modified": "2014-01-27 16:11:24.314194+00",  
"label": null, "owner": "u:blaiszik", "writeok": true, "id": 49, "modified by": "u:blaiszik"}]
```

8. View your Catalog via the UI

- a. The UI is located at: <https://catalog-alpha.globuscs.info/>



CLI Related Files:

catalog-client/globusonline/catalog/client/examples/catalog.py
catalog-client/globusonline/catalog/client/examples/catalog_wrapper.py

catalog-client/globusonline/catalog/client/examples/test_command_line.sh

Web GUI: <https://catalog-alpha.globuscs.info/datasets>

Authentication: Use your account from the test GO instance at <http://test.globuscs.info>

Flags

The flag must be positioned immediately after the catalog.py command.
The flag may be given in the single letter format or the long format.

- **-h --help**
Print a help message summarizing the options
- **-f --force**
Ignore non-critical errors. See commands for details.
- **-t --text**
Receive formatted text output instead of JSON
- **-dc** **THIS FLAG DOES NOT EXIST: use of default catalog is determined just by presence of the GCAT_DEFAULT_CATALOG_ID env var.**
Add the -dc flag to the command to retrieve a default catalog id from the local environment variable GCAT_DEFAULT_CATALOG_ID. When using this flag, remove CATALOG_ID from any command line arguments listed below
*the exception is delete_catalog, which does not accept a default catalog value
*All commands with an **[&]** in the title currently accept default_catalog as an argument
- **-n --name**
Operate on names instead of dataset IDs
The number of matching entries must be 1 or an error occurs
- **-s --short**
Generate plain text output in short format - suitable for shell scripting
- **-x:** Suppresses all output from the CLI

Environment Variables

- Setting GCAT_DEFAULT_CATALOG_ID sets the default catalog without requiring the -dc flag

Example: `python catalog.py -t get_catalogs`
`python catalog.py -t -dc get_datasets`
 *Assuming you have GCAT_DEFAULT_CATALOG_ID environment variable set

Command Examples

Get all Catalogs

Returns a JSON formatted string of the catalogs

Usage/Example: `python catalog.py get_catalogs`

Create Catalog

Usage: `python catalog.py create_catalog <Property JSON>`

Examples:

```
python catalog.py create_catalog '{"config": {"name": "Test Catalog"}}'
python catalog.py create_catalog 'Test Catalog'
```

Delete Catalog

Usage: `python catalog.py delete_catalog CATALOG_ID 'true'`

Example:

```
python catalog.py delete_catalog 10 'true'
```

Create Dataset [&]

Usage: `python catalog.py create_dataset CATALOG_ID <Dataset JSON>`

Examples:

```
python catalog.py create_dataset 17 '{"name":"New Dataset"}'  
python catalog.py create_dataset 'New Dataset'
```

Get all Datasets [&]

Usage: `python catalog.py get_datasets CATALOG_ID`

Examples:

```
python catalog.py get_datasets 17  
python catalog.py get_datasets
```

Notes: Supports text and short format

Delete Dataset [&]

Usage: `python catalog.py delete_dataset CATALOG_ID DATASET_ID 'true'`

Examples:

```
python catalog.py delete_dataset 17 65 'true'  
python catalog.py -n delete_dataset 17 'New Dataset' 'true'  
python catalog.py -n delete_dataset 'New Dataset' 'true'
```

Notes: Supports -n

Get all Dataset Members [&]

Usage: `python catalog.py get_dataset_members CATALOG_ID DATASET_ID`

Examples:

```
python catalog.py get_dataset_members 17 54  
python catalog.py -n get_dataset_members 17 'New Dataset'  
python catalog.py -n get_dataset_members 'New Dataset'
```

Notes: Supports -t and -s and -n

Get Catalog Annotation Definitions [&]

Usage: `python catalog.py get_annotation_defs CATALOG_ID`

Examples:

```
python catalog.py get_annotation_defs 17  
python catalog.py get_annotation_defs  
python catalog.py -t get_annotation_defs
```

Notes: Supports -t

Get Dataset Annotations [&]

Usage: `python catalog.py get_dataset_annotations CATALOG_ID DATASET_ID`

Examples:

```
python catalog.py get_dataset_annotations 17 100
python catalog.py -n get_dataset_annotations 17 'New Dataset'
python catalog.py -n get_dataset_annotations 'New Dataset'
```

Notes: Supports -t and -n

Create Annotation Definition [&]

Usage: `python catalog.py create_annotation_def CATALOG_ID ANNOTATION_NAME ANNOTATION_TYPE MULTI_VALUED(optional)`

Examples:

```
python catalog.py create_annotation_def 48 'myIntegerAnnotation' 'int8'
python catalog.py create_annotation_def 48 'myTextAnnotation' 'text' True
```

* Second command creates a multivalued annotation def

* Available Annotation types {enum: ['text', 'int8', 'float8', 'boolean', 'timestampz', 'date']}

Notes: Results in an error if annotation is already defined. Use -f to ignore this error.

Add Dataset Annotation [&]

Usage: `python catalog.py add_dataset_annotation CATALOG_ID DATASET_ID <Annotations>`

Example:

```
python catalog.py add_dataset_annotation 17 54 '{"nexus":"false"}'
python catalog.py add_dataset_annotation 54 myNewAnnotation:'xyz'
python catalog.py add_dataset_annotation 54 a:'123' material:'copper'
python catalog.py -n add_member_annotation "New Dataset"
test-annotation:true material:copper
python catalog.py -n add_member_annotation 'New Dataset' test-annotation:true
material:copper
```

**Note that before adding annotations, you must add an annotation definition if the type is not text*

Notes: Supports -n

Query Dataset [&]

Usage: `python catalog.py query_datasets CATALOG_ID FIELD OPERATOR VALUE`

Examples:

```
python catalog.py query_datasets 17 name LIKE %New%
```

*returns all datasets from catalog 17 with name matching %New% where % is a wildcard

```
python catalog.py query_datasets 17 id GT 100
```

*returns all datasets in catalog 17 with ID greater than 100

```
python catalog.py query_datasets material LIKE %Azo%
```

*returns all datasets where the 'material' annotation is LIKE %Azo% (i.e. match xyzAzoxyz)

```
python catalog.py query_datasets 17 material LIKE %
```

*returns all datasets with the material tag set to any value

Notes: Supports -t and -s

Operators (explanation) USE CAPS portion in query:

EQUAL, NOT_EQUAL, GT (greater than), GEQ (greater than or equal), LT (less than), LEQ (less than or equal), FULLTEXT, NOT_FULLTEXT, LIKE, SIMTO, REGEXP, NOT_REGEXP, REGEXP_CASE_INSENSITIVE, NOT_REGEXP_CASE_INSENSITIVE

The following non-alphabetic operators are aliases: == != > >= < <=

Either of these may be used for LIKE: ~ ~~

Example:

```
python catalog.py query_datasets 17 id '>' 100
```

Create Dataset Members [&]

Usage: `python catalog.py create_members CATALOG_ID DATASET_ID <Member JSON>`

Output: New dataset ID.

Examples:

```
python catalog.py create_members 17 86 '{"data_type":"file",  
"data_uri":"globus://go#ep1/~/.test.tst"}'  
python catalog.py create_members 17 86 file 'globus://go#ep1/~/.test.tst'
```

*This will automatically add the appropriate data_reference to the member

Notes: Supports -n, as in:

```
python catalog.py -n create_members 17 'New Dataset' file  
'globus://go#ep1/~/.test.tst'
```

Delete Dataset Annotation [&]

Usage: `python catalog.py delete_dataset_annotation CATALOG_ID DATASET_ID
ANNOTATION_KEY ANNOTATION_VALUE`

```
python catalog.py delete_dataset_annotation 17 54 nexus false
```

Add Member Annotation [&]

Usage: `python catalog.py add_member_annotation CATALOG_ID DATASET_ID MEMBER_ID
<Annotations>`

Example:

```
python catalog.py add_member_annotation 17 54 86 '{"nexus":"false"}'  
python catalog.py add_member_annotation 17 54 86  
 '{"test-annotation":"true","material":"copper"}'  
python catalog.py add_member_annotation 17 54 86 test-annotation:true  
material:copper  
python catalog.py -n add_member_annotation 17 "New Dataset" 86  
test-annotation:true material:copper
```

*Note that before adding annotations, you must add an annotation definition

Annotations may be in JSON format or plain `key:value` format

Notes: Supports -n

Delete Member Annotation [&]

Usage: `python catalog.py delete_member_annotation CATALOG_ID DATASET_ID MEMBER_ID ANNOTATION_KEY ANNOTATION_VALUE`

`python catalog.py delete_member_annotation 17 54 86 nexus false`

Add Dataset Acl

Add Dataset Acl Recursive

Get Dataset Acl

Usage: `python catalog.py get_dataset_acl CATALOG_ID DATASET_ID`

Example:

`python catalog.py get_dataset_acl 17 54`

Delete Token File

Allows you to remove your token file if it currently exists

Usage/Example: `python catalog.py delete_token_file`

Create Token File

Allows you to create and store your user authentication token. Note that this is done automatically with other function calls.

Usage/Example: `python catalog.py create_token_file`

Known Bugs:

- **Add bugs here**

Feature Request:

- **Automatically add a default annotation def if it does not exist?**
 - Default type is text
 - An ability to define annotations and add them on the fly
- **Export as a pdf in chrono format +++**
 - Web GUI interface for PDF export?
- **Response code for errors +++**
- **Error handling strategy +++**
 - Push to a local log file and recover
 - Execution log file (Suresh)