*HPC Data MANAGEMENT*

User guide

Version *1.6*

*12/26/2017*

TABLE OF CONTENTS

[1. Purpose 5](#_Toc503274510)

[2. Introduction 5](#_Toc503274511)

[2.1 What is HPC DME ? 5](#_Toc503274512)

[2.2 Intended users 5](#_Toc503274513)

[2.3 Accessing HPC DME 5](#_Toc503274514)

[3 Pre-Requisites 6](#_Toc503274515)

[3.1 Account with HPC DME 6](#_Toc503274516)

[3.2 Using Globus for asynchronous data transfers 6](#_Toc503274517)

[3.3 Pre-requisites for HPC DME utilities and CLI 8](#_Toc503274518)

[3.3.1 Java 8](#_Toc503274519)

[3.3.2 Get the HPC DME CLI 9](#_Toc503274520)

[4 HPC DME Overwiew 9](#_Toc503274521)

[4.1 Business rules and characteristics of collections 10](#_Toc503274522)

[4.2 Business rules and characteristics of data objects 11](#_Toc503274523)

[4.3 Metadata 12](#_Toc503274524)

[4.4 User authentication 13](#_Toc503274525)

[5 HPC DME Core FUNCTIONALITIES 13](#_Toc503274526)

[5.1 Group administrator functions 14](#_Toc503274527)

[5.1.1 Enroll user into HPC DME (create) 14](#_Toc503274528)

[5.1.2 Add or remove user(s) from a group 14](#_Toc503274529)

[5.1.3 Delete data object by group administrators only 14](#_Toc503274530)

[5.2 Register/Upload functions 15](#_Toc503274531)

[5.2.1 Register collection with its metadata into HPC DME(create and update) 15](#_Toc503274532)

[5.2.2 Register data object along with metadata into HPC DME (create and update) 15](#_Toc503274533)

[5.3 Assign permissions on HPC DME collections or data objects 15](#_Toc503274534)

[5.4 Search functions 15](#_Toc503274535)

[5.4.1 Find collection by metadata 15](#_Toc503274536)

[5.4.2 Find data object by metadata 16](#_Toc503274537)

[5.5 Get collection by path 17](#_Toc503274538)

[5.6 Get data object by path 18](#_Toc503274539)

[5.7 Download functions 18](#_Toc503274540)

[5.7.1 Download data object/file to another globus endpoint 18](#_Toc503274541)

[5.7.2 Download data object/file to local file system 18](#_Toc503274542)

[6 Executing HPC DME API with Command Line Client 18](#_Toc503274543)

[6.1 Seting the HPC DME CLI 19](#_Toc503274544)

[6.2 Running HPC DME Command line Client 20](#_Toc503274545)

[6.2.1 Interactive Mode: 20](#_Toc503274546)

[6.2.2 Batch Mode 21](#_Toc503274547)

[6.3 Error handling 33](#_Toc503274548)

[7 Executing HPC DME API with Command line utilities 33](#_Toc503274549)

[7.1 One time setup 33](#_Toc503274550)

[7.2 Generate a Data management API token 33](#_Toc503274551)

[7.3 Generate a Globus token 34](#_Toc503274552)

[7.4 Register a collection 34](#_Toc503274553)

[7.5 Register a dataobject 34](#_Toc503274554)

[7.5.1 Synchronously (From file system) 34](#_Toc503274555)

[7.5.2 Asynchronously (From Globus) 35](#_Toc503274556)

[7.5 Register a Directory 35](#_Toc503274557)

[7.5.1 Synchronously (from file system) 35](#_Toc503274558)

[7.5.2 Asynchronously (from Globus) 36](#_Toc503274559)

[7.6 Get a collection metadata 36](#_Toc503274560)

[7.7 Get a dataobject metadata 36](#_Toc503274561)

[7.8 Download a dataobject synchronously 36](#_Toc503274562)

[7.9 Download a Collection or dataobject Asynchronously 36](#_Toc503274563)

[8 Executing HPC DME API with CURL 37](#_Toc503274564)

[8.1 Setting up the environment 37](#_Toc503274565)

[8.2 Executing CURL commands 37](#_Toc503274566)

[8.2.1 Registering a new user 37](#_Toc503274567)

[8.2.2 Registering a collection 38](#_Toc503274568)

[8.2.3 Registering a data object 38](#_Toc503274569)

[8.2.3.1 Registering a data object from a Globus endpoint 38](#_Toc503274570)

[8.2.3.2 Registering a data object from the file system 38](#_Toc503274571)

[8.2.4 Search for collection 38](#_Toc503274572)

[8.2.5 Search for data Object 39](#_Toc503274573)

[8.2.6 Assign Permissions 39](#_Toc503274574)

[8.2.7 Download data object to a Globus endpoint 39](#_Toc503274575)

[8.2.8 Download data object to a file system 39](#_Toc503274576)

[8.2.9 Delete Data Object (by group administrators only) 40](#_Toc503274577)

[9 Executing HPC DME API with Web Client 40](#_Toc503274578)

[9.1 Dashboard and login/logout 40](#_Toc503274579)

[9.1.1 Log In 40](#_Toc503274580)

[9.1.2 Log Out 40](#_Toc503274581)

[9.1.3 Dashboard 41](#_Toc503274582)

[9.2 Search functions 41](#_Toc503274583)

[9.2.1 Google Style Search 42](#_Toc503274584)

[9.2.2 Basic Search 42](#_Toc503274585)

[9.2.3 Advanced Search 42](#_Toc503274586)

[9.2.4 Saved Search 42](#_Toc503274587)

[9.2.5 Bookmark Function 43](#_Toc503274588)

[9.3 BROWSE functions 45](#_Toc503274589)

[9.4 Register a collection 47](#_Toc503274590)

[9.5 Register a Data File 50](#_Toc503274591)

[9.6 SHARING functions 52](#_Toc503274592)

[9.7 Update metadata in non-batch mode 54](#_Toc503274593)

[9.8 Download Collection/file from archive 55](#_Toc503274594)

[9.8.1 Share a Globus End Point with HPCDME Application Account 55](#_Toc503274595)

[9.8.2 Download a Collection of Data Files 56](#_Toc503274596)

[9.8.3 Download a Single Data File 57](#_Toc503274597)

[9.9 Notifications and reports 59](#_Toc503274598)

[9.9.1 Notifications Management 59](#_Toc503274599)

[9.9.2 Reports Management 59](#_Toc503274600)

[9.10 Group administrator functions 60](#_Toc503274601)

[9.10.1 User 61](#_Toc503274602)

[9.10.2 Group 62](#_Toc503274603)

[9.10.3 Delete a Data Object and Associated Metadata 64](#_Toc503274604)

[10 APPENDIX A – sample collection input 66](#_Toc503274605)

[11 APPENDIX B – sample data file input 66](#_Toc503274606)

[12 Appendix c – sample permissions input 66](#_Toc503274607)

[13 APPENDIX D – Executing HPC DME API with Soap UI 66](#_Toc503274608)

[13.1 Prerequisites 66](#_Toc503274609)

[13.2 Access HPC DME with Soap UI 68](#_Toc503274610)

[13.3 Use SOAP UI to perform common use scenarios 68](#_Toc503274611)

[13.3.1 Register a collection (Project or dataset) 68](#_Toc503274612)

[13.3.2 Register a single data file/object into storage archive asynchronously 70](#_Toc503274613)

[13.3.3 Register a single data file/object into storage archive synchronously 72](#_Toc503274614)

[13.3.4 Perform update on a metadata attribute 73](#_Toc503274615)

[13.3.5 Subscribe to a known event 74](#_Toc503274616)

[13.3.6 Generate a report 75](#_Toc503274617)

[13.3.7 Update/assign permission 76](#_Toc503274618)

[13.3.8 Perform simple search functions 76](#_Toc503274619)

[13.3.9 Download a data file/object to Globus share 79](#_Toc503274620)

[13.3.10 Download a data file/object to a local directory 80](#_Toc503274621)

[14 APPENDIX E HPC DME CLI properties 80](#_Toc503274622)

# Purpose

The user guide describes the core High Performance Computing (HPC) Data Management Environment (DME) functional capabilities, i.e., registering (uploading), transferring, sharing, searching, downloading collections (i.e., datasets, projects, or folders) and data objects (i.e. files). It further outlines various means of performing such actions from an end user angle: i.e., executing core APIs from Web, command line client and utilities, and CURL. The user guide also includes details of the prerequisites needed for setting up the HPC DME clients**.**

# Introduction

## What is HPC DME ?

The HPC DME, High Performance Computing Data Management Environment, is a highly adaptable and an open ended data storage environment supporting storage and management of data, produced from high performance computing systems. HPC DME provides capabilities for storing, managing, transferring and sharing these data across different storage systems securely and efficiently.

Users can store data objects for a long term on HPC DME, share and transfer their data such that they do not have to redistribute or maintain copies of the data on other systems by eliminating the data integrity issues. HPC DME stores and associates user defined metadata to any registered data at different levels of data life cycle, enabling the environment not only to help identify the data but also enhancing the search capabilities and to be able to attach a value factor to each dataset.

## Intended users

The HPC DME has been built to cater to the data storage and data management needs of NCI cancer community. Any user with a valid HPC DME user account can run its client interface. *Note: HPC DME supports only NIH Active Directory account holders at this point of time*.

## Accessing HPC DME

The HPC DME APIs and associated servers and infrastructures are deployed at Development, UAT and Production tiers for ongoing development, user acceptance and exploration usage with limited storage allocated and CleverSafe vault customized through the Frederick ITOG team. Data storage need will be assessed, estimated and monitored as a new user or group of users want to join for this pilot use.

HPC DME can be accessed through its Service APIs. HPC DME Service APIs are developed based on REST standard. Therefore, these APIs can be accessed by any interface that is compatible with REST standard. For example, these Service APIs can be accessed through Java, Perl, Ruby, Curl, JavaScript languages programmatically or through UI clients like SOAP UI, REST Client (See [Appendix D: Using Soap UI to execute HPC DME APIs](#_APPENDIX_D_-)) .

We have also provided an example implementation to access these HPC DME APIs using a web GUI, and command line utilities for shell users. Detailed information on accessing these service APIs will be the focus of this User Guide.

The following is the HPC DME Service API URL for the production tier:

<https://hpcdmeapi.nci.nih.gov:8080>

The following is the HPC DME Service API URL for the User Acceptance Test (UAT) tier:

<https://fr-s-hpcdm-uat-p.ncifcrf.gov:7738/hpc-server>

# Pre-Requisites

## Account with HPC DME

You will first need to register with HPC DME for an account. Please consult HPC DME administrator for an account at [HPC\_DME\_Admin@nih.gov](mailto:HPC_DME_Admin@nih.gov) after obtaining your group or DOC approval of using the HPC DME. If you are a HPC DME administrator, please see HPC Server API specification for user registration details.

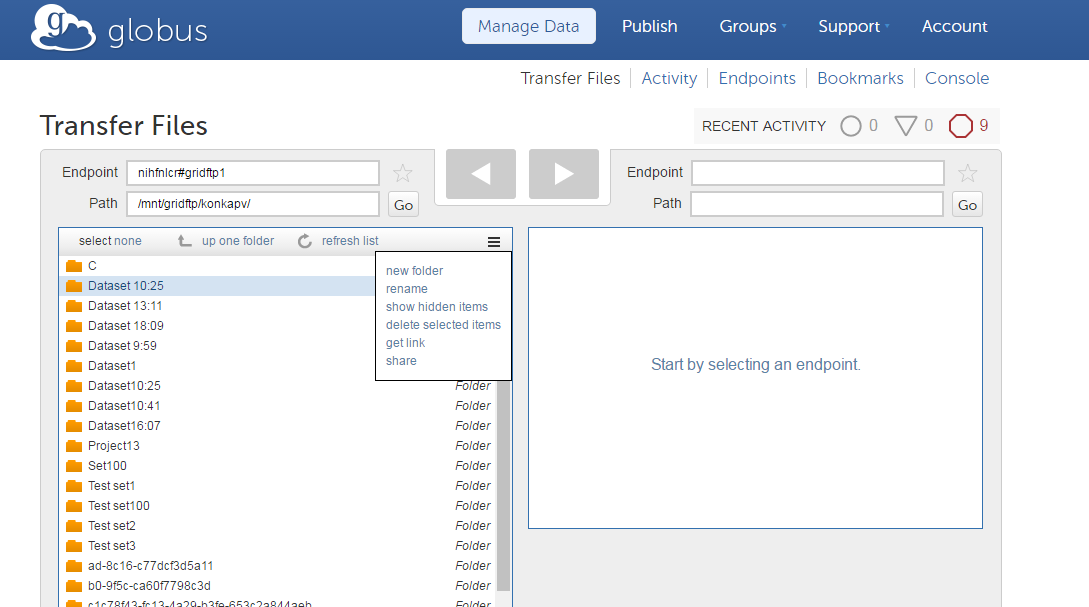
## ­Using Globus for asynchronous data transfers

HPC DME supports transferring data from a Globus endpoint to HPC DME Archive storage, and from HPC DME Archive storage to a Globus endpoint.

Though using Globus is not necessarily a prerequisite to transfer data files (objects) into and from the Cleversafe object store or any other storage medium like isilon storage system, it is recommended for transferring large data objects (greater than or equal to 5 GB) asynchronously.

The HPC DME environment uses a Globus application account to perform asynchronously transfer from and to Cleversafe. If you would like to register data that resides on a Globus endpoint into the archive, you should first grant READ permission to the application account before your registration. Similarly, if you would like to download data from the archive to a Globus endpoint, you should grant the application account a WRITE permission to the shared Globus Endpoint. Note that sharing a Globus endpoint is only available if you have Globus Plus account. Biowulf, Frederick, and CBIIT endpoints have Globus Plus accounts, while the Globus connect personal endpoint does not have this privilege by default. For more information about sharing a Globus endpoint and granting access permission, please review [How To Share Data Using Globus](https://docs.globus.org/how-to/share-files/). You can also contact [staff@hpc.nih.gov](mailto:staff@hpc.nih.gov) if you would like to have a Globus Plus account on your server or laptop.

Globus do not support sharing individual files. If you have a single file to register, please create a folder for it and share the folder.



Here are the application accounts for different HPC DME tiers:

|  |  |  |
| --- | --- | --- |
| Globus Application Account | Globus Application ID String | Applicable HPC DME Tier (s) |
| HPCDME-DEV-APP-ACCOUNT | 025ff462-07e1-483b-8dbb-1fc26c7eb17e@clients.auth.globus.org | Development/UAT |
| HPCDME-PROD-APP-ACCOUNT | 07cec4f4-8d99-4b9d-a513-d9a1148ac054@clients.auth.globus.org | Production only |

If you are planning to transfer large number or size of files from or to Globus Endpoints you have access to, we provide a command line utilities to facilitate this bulk data transfer operation. This command line client would need you to create a Globus ID and link it to your NIH AD account. This way you will be able to authenticate on Globus using your Globus password from command line, instead of using your NCI password on iTrust.

To create a Globus Id:

1. Go to <https://www.globusid.org>
2. Click on “Create a globus ID”
3. Follow the instructions

To link your Globus ID with your NIH account:

1. Login into [www.globus.org](http://www.globus.org) with your Globus Id
2. Click on “Account” button which is at the top right corner. Click on “Add Linked Identity”
3. Select “National Institutes of Health” from the identity provider list box. And click on Continue button.
4. Now you will be redirected to NIH iTrust Website. Enter your NIH AD credentials and submit.
5. If you are asked to enter your Globus Credentials, enter those and click continue.
6. Now Globus and NIH accounts are linked. Note: Globus does not track or keep NIH credentials but use NIH identity provider to authenticate your NIH credentials.

## Pre-requisites for HPC DME utilities and CLI

### Java

Download and install Java 1.8.x

<https://www.java.com/en/download/>

Please visit the following link for installation instructions

<https://www.java.com/en/download/help/index_installing.xml>

After successful installation of Java, update PATH with Java executable folder (Ex: c:/jdk1.8\_13/bin)

<https://java.com/en/download/help/path.xml>

Open cmd/shell window and type “java –version” to make sure it displays the version information you installed.

### Get the HPC DME CLI

Clone HPC DME Utils into your local machine.

Git clone <https://github.com/CBIIT/HPC_DME_APIs>

Navigate to HPC\_DME\_APIs/utils folder. Open README.md and follow the instructions.

Note: For windows users, you have to first install [Cygwin](https://www.cygwin.com/) to use the command line utilities. You can also open a terminal cmd to run the java HPC DME CLI.

# HPC DME Overwiew

The HPC data management environment provides a high-reliability storage model for underlying collections including a collection registration system, and an API for transfer of data objects. The data object registration associates a label with a given managed data file or folder and captures extensible metadata for the managed data object.

The HPC data management environment provides a number of application programming interfaces (APIs) to operate and interact with it. At a high level, there are two important components in HPC DME:

1) Metadata management: HPC DME by default integrates with iRODS iCAT instance to manage metadata and its security for both collections and data objects.

2) Data transfer: HPC DME can transfer data directly from a file system (synchronous registration), or use Globus to perform asynchronous data transfer between Globus endpoint and CleverSafe storage (asynchronous registration).

Synchronous registration integrates with S3 API to transfer data from local file system to HPC DME archive. The current approach is to stream data files (objects) from local/designated file directory to the application server, and then stream that to the Cleversafe archive.

HPC DME pluggable architecture allows both these implementations to be replaced with alternatives easily while keeping its APIs unchanged. The basic features of HPC DME is to help users in registering and uploading their data to the HPC DME archive storage and managing it. HPC DME archive storage can be a permanent storage for the users’ data and be used as a platform to search, manage and transfer the data to other storage systems and also to share with other collaborators or users. Each data object is stored along with its required and user defined metadata associated with it. The associated metadata can be used as search criteria to identify dataset(s).

The HPC data management implementation provides users flexibility to define collections and data objects. Collections in HPC DME can be Projects, Datasets, etc. and data objects are the data files. Data object is defined as an individual file. Collections can be of different types for example, a dataset can be linked to a sequencing project, imaging study or an analysis project etc., or a project can be an entity (umbrella project) indicating a scientific ‘study’, or simply a folder. To summarize, HPC DME supports metadata registration to connect unstructured data files or objects. Metadata may be attached to files, collections (Equivalent of sub-directories). Though each laboratory or NCI business unit may adopt or engage in a variety of biomedical/clinical research and produce many forms of data, HPC DME does enable flexibility for each DOC to have their own way of organizing/structuring how their data/metadata is presented or preserved.

The HPC DME database will store metadata in the form of “triples”. The triples consist of an attribute field, a value field, and a unit field. The content of each of these fields can be independently defined and applied. Metadata may be user-defined or applied programmatically.

## Business rules and characteristics of collections

* Each collection can be referred with a logical path (“collection\_path”) which is unique across HPC DME.
* Irrespective of the logical path, each collection is associated with UUID metadata attribute to uniquely identify itself.
* A collection can be any category of business domain entities such as: Project, Dataset, Folder, Samples, Runs. More types can be configured in the system easily through metadata configuration. Please consult your HPC DME system administrator.
* Users can create/register multiple collections with HPC DME.
* Each collection can have its own required set of metadata which needs to be submitted at the time of registration. This required metadata can be configured by a system administrator with input from each DOC’s group administrator.
* Users have an option of adding new metadata variables to the required metadata associated with each collection.
* A collection can contain one or more data objects, or child collections. This is achieved using the logical path of a given collection.

## Business rules and characteristics of data objects

* A data object can be a single data file.
* Each data object can be referred with a logical path (“object\_path”) which is unique across HPC DME.
* A parent collection should be registered first with HPC DME before registering a data object.
* Irrespective of the logical path, each data object is associated with UUID metadata attribute to uniquely identify itself.
* Each data object can have associated required metadata which must be submitted at the time of data object registration.
* Users have an option of adding new metadata variables to the required metadata associated with each data object.

To emphasize, HPC DME supports DOC-dependent organizations of their biomedical data/metadata, user defined collection types, more importantly allows and fully supports specific validation rules (policies) when they are registered into the archive. It is up to each DOC to come up, agree upon and determine how these policies will be enforced, standardized, to certain extent. Some DOC groups may script or automate the generations of their actual data/metadata files in their LIMS system to enable seamless integration between LIMS and HPC DME with little human intervention, verifying against all rules defined in the policy file.

## Metadata

Metadata is defined as the data about the data. It is the information describing the actual data such as the date and origin of creation, its contents, its condition, its processing, or its associations to other objects. Metadata is employed to make data searches faster, more specific and also enable and promote data sharing among scientists.

HPC DME collects metadata for each collection and data object, registered and stored in a database along with the associations. HPC DME collects two kinds of metadata related to a collection or a data object, namely, administrative and center/division specific. The administrative metadata is the required set of information which needs to be submitted at the time of registration with HPC DME. New metadata variables can be added to both administrative and center/division specific metadata sets after a user obtains proper authorizations and permissions. The metadata can also be updated by authorized users through API or command line client.

HPC DME supports user defined collection types. Some of the collections can be Project, Sample, Study, Dataset. Collections types should be defined in the system policy file to enforce validation rules. To create new collection types, please submit the following information to HPC DME System Administrator.

Collection type name and its metadata in the following format:

{

"attribute":"<Name of the attribute>",

"mandatory":<true/false>,

"defaultValue":"<Value>",

"collectionTypes":<Collection Type name>,

"validValues":[

"<Value1>",

"<Value2>"

],

"ruleEnabled":true,

"DOC":"<DOC-NAME>"

}

## User authentication

After enrolling with HPC DME, the users can login into HPC DME using their user id. The HPC DME validates the user passwords through the NIH AD account.

# HPC DME Core FUNCTIONALITIES

HPC DME Service API is a RESTful interface. This interface is communicated over HTTPS with the same HTTP verbs (GET, POST, PUT, DELETE) that web browsers use to retrieve web pages and to send data to remote servers. HPC DME Service API can be categorized into the following sections. Please see HPC DME Server API specifications for details on each of these functions.

HPC DME APIs are developed on REST standard, so any client interface can be used to access these APIs confirming REST standards. In this user guide, we provide a web interface and command line utilities to access the HPC DME APIs. The web interface is intended for users who are not familiar with command line. The command line utilities are intended for users who are familiar with the shell and consists of: a) The HPC DME CLI which is a java package for bulk processing and permission assignment on Linux and windows users. b) The command line utilities are used as wrapper for most common register and download operations. c) The curl utilities give examples that show how to call the HPC DME API if the user need finer control over calling the APIs.

In a typical scenario, consumer users with the need of searching, browsing, sharing, and downloading data, will use the Web GUI. Group administrators, as the data owner for a DOC, will also use the Web GUI to manage their groups and enroll users. The command line utilities will be mostly used by data producers to archive bulk amount of data.

All these will be extended further below to detail how end users may use various clients, specifically the Web GUI, to perform needed and supported operation for managing archived data/metadata of their interest. If you are planning to use Globus for data transfer, please refer to section [3.2 on how to setup your Globus account.](#_Globus_Account_for)

## Group administrator functions

## [Enroll user into HPC DME](#_Enroll_User) (create)

[Enroll a User via Web GUI Client](#_User)

[Enroll a User via CURL](#_Registering_a_new)

## Add or remove user(s) from a group

[Add or remove user(s) from a group via Web GUI Client](#_Group)

## Delete data object by group administrators only

*This delete data object and associated metadata function is implemented to support rare case scenarios where an end user may make mistakes in registering the wrong files. In consulting the users, only a group administrator with a data ownership to the object file will be allowed to perform this function to enforce and maintain data integrity. Moreover, associated metadata will be deleted from the database.*

[Delete Data Object via Web GUI Client](#_Delete_a_Data)

## Register/Upload functions

## [Register collection with its metadata into HPC DME](#_Register_Project)(create and update)

[Register Collection via Web GUI](#_Register_a_collection)

[Register Collection via Command Line Client](#_Register_collections)

[Register Collection via Command Line Utility](#_7.4_Register_a)

[Register a Directory via Command Line Utility](#_Register_a_Directory)

[Register Collection via CURL](#_Registering_a_collection)

## [Register data object along with metadata into HPC DME](#_Register_Dataset) (create and update)

In case of asynchronous dataObject registration using Globus, you have to provide the endpoint Unique Universal ID UUID to the API. You can retrieve a Globus endpoint UUID by clicking on “Endpoints” tab link in the Globus portal (after login) and click on the endpoint name. You should see “UUID” if you scroll down in the “Overview” section. Use UUID value for “fileContainerId”.

[Register a Data File via Web GUI](#_Register_a_Data)

[Register data object (Create & Update) via Command Line Utility](#_7.5_Register_a)

[Register a Directory via Command Line Utility](#_Register_a_Directory)

[Register data object (Create & Update) via Command Line Client](#_Registering_Data_Objects_1)

[Register data object (Create & Update) via CURL](#_Registering_a_data)

## Assign permissions on HPC DME collections or data objects

[Sharing/Authorization via Web GUI Client](#_SHARING_functions)

[Sharing/Authorization via Command Line Client](#_Update_permissions)

[Sharing/Authorization via CURL](#_Assign_Permissions)

## Search functions

## [Find collection by metadata](#_Find_Project_by_1)

[Find Collection by path via Web GUI Client](#_Search_functions)

[Find Collection by metadata via Command Line Client](#_Search_Collections_by)

[Find Collection by metadata via CURL](#_Search_for_collection)

## [Find data object by metadata](#_Find_Dataset_by_2)

[Find Data Object by metadata via Web GUI Client](#_Search_functions)

[Find Data Object by Metadata via Command Line Client](#_Search_Data_objects)

[Find Data Object by Metadata via CURL](#_Search_for_data)

The default hierarchical search APIs is based on the fact that every collection and dataObject inherits the metadata of all its parent collections. However, there should be no assumption made on the uniqueness of metadata attribute variables across a given hierarchy.

Assume a registered file in the following path: /Coll\_A/Coll\_B/Coll\_C/File\_D, it is possible that the same metadata variable “attribute\_X” be used for “Coll\_C” and “File\_D”. In this case, the value assigned to this attribute at the lower level (i.e., File\_D) will overwrite the value assigned to this attribute at the parent level (i.e., Coll\_C) unless otherwise specified as explained next.

Default hierarchical search using compound query (please review the HPC\_server\_API document) will search the data object and collection at all levels (>=1) if no level value is indicated by the users. The lowest level in the hierarchy is given the value ‘1’. During the hierarchical search, users can also indicate the optional level they are basing the search on. For example, while searching for a dataObject, the dataObject level will equal ‘1’, the parent collection level will equal ‘2’, the grandparent collection level will equal ‘3’, etc.

Example 1: Given the path of File\_D shown above, assume you attached metadata attributes as the following:

* Coll\_A:  “X” = “1” (i.e., Metadata attribute name “X” set to a value of “1”)
* Coll\_B: “Y” = “2”
* Coll\_C: “Z” = “3”
* File\_D: “W” = “4”

Since there are unique metadata variables, there would be no difference made whether or not the users enter the level parameter values in entering the search criteria for the attribute “X”, “Y”, “Z”, or “W” because it will search at all levels.

Example 2: Assume you attached Metadata attributes as the following:

* Coll\_A: “X” = “1”
* Coll\_B: “Y” = “2”
* Coll\_C: “date\_created” = “11-11-2016”
* File\_D: “date\_created” = “11-20-2016”

Since “date created” may be assigned different values for the parent collection “Coll\_C” and the child “File\_D”, it is critical for the users to indicate at what level they are performing the search. However, if no level is indicated in the entered search criteria, default search will always search at all levels (collections and data files). If the user wants to search for a dataObject whose parent collection was created at a specific date, the compound query should include a search for the attribute “date\_created” with the value “11-11-2016” at level 2, with the level operator “EQUAL”. Therefore, the level operator specifies where to apply the search criteria.

Note that the default search level for data files is at all collection and data object levels. However, if a search is specifically indicated for collections, the default search is at all collection levels. In other words, while searching for collection, the search criteria will be executed at all collection levels (except data object level 1). The rationale is that there is nothing harmful to show a broader collections result set regardless if any data objects have been registered under certain collection path.

## Get collection by path

[Find Collection by path via Web GUI Client](#_Search_functions)

[Find Collection by path via the Command Line Utility](#_Get_a_collection)

[Find Collection by path via Command Line Client](#_Search_Collection_by)

[Find Collection by path via CURL](#_Search_for_collection)

## Get data object by path

[Find Data Object by Path via Web GUI Client](#_Search_functions)

[Find Data Object by Path via Command Line Utility](#_get_a_dataobject)

[Find Data Object by Path via Command Line Client](#_Search_Data_file)

[Find Data Object by Path via CURL](#_Search_for_data)

## Download functions

## Download data object/file to another globus endpoint

First setup your Globus account as mentioned [here](#_Using_Globus_for_1).

[Download data objects to Globus endpoint via Web GUI Client](#_Download_Collection/file_from)

[Download data objects to Globus endpoint via CMD Line Utility](#_Download_a_Collection)

[Download data objects to Globus endpoint via CURL](#_Download_data_object)

## Download data object/file to local file system

[Download data objects to Local File System via Web GUI Client](#_Download_Collection/file_from)

[Download data objects to Local File System via CMD Line Utility](#_Download_a_dataobject)

[Download data objects to Local File System via CURL](#_Download_data_object_1)

# Executing HPC DME API with Command Line Client

The HPC DME command line client is an interactive tool where users can enter commands to initiate supported functions. Current version of HPC client supports the following functions:

* Batch registeration of collections and their associated metadata
* Batch register of data objects and their associated metadata
* Batch registration of local directories

Recursively register all files from a given local directory on your file system with HPC DME. This register each directory and file with basic metadata and optional custom metadata. Inclusion and exclusion criterion can be set in include or exclude specific file and directory patterns.

* Batch registration of files from a Globus Endpoint

Recursively register all files from a given Globus endpoint location with HPC DME. This register each file with basic metadata.

* Update user permissions on accessing HPC DME metadata
* Search for collections based on search criteria and write results into a csv/json/txt file
* Search for data objects based on search criteria and write results into a csv/json/txt file

HPC Client is integrated with HPC Server REST interfaces to perform the functions mentioned above. For batch registration of collection and dataObjects, it takes CSV (comma separated value) file as input for the commands it supports. HPC client parses given CSV file input and invokes server APIs in a batch manner to perform requested commands.

HPC Client commands are secured with valid authentication. To run protected commands, a user would need to provide HPC credentials to get authenticated. All client commands do not support multi-thread processing of input requests at this time.

## Seting the HPC DME CLI

Follow [these instructions](#_Get_the_HPC) to get the HPC DME CLI.

Before running a HPC client, you would need to update HPC client properties file /PATH/TO/HPC\_DME\_APIs/utils/hpcdme.properties

The following are the important properties:

|  |  |  |
| --- | --- | --- |
| Property name | Description | Default value |
| hpc.server.url | HPC Server API url. This is where REST interfaces are running on the server | Please see [section 2.3](#_Accessing_HPC_DME) for environment specific API URL |
| hpc.ssl.keystore.path | Path of the keystore used for 2-way SSL connectivity with HPC Server. If the value is missing, one way SSL connection is used. | -For UAT server:  hpc-client/keystore/keystore-uat.jks  -For dev server:  hpc-client/keystore/keystore-dev.jks |

Please see this [APPENDIX E](#_APPENDIX_E_HPC) for the full list of properties

## Running HPC DME Command line Client

HPC client is a command line tool that can be run using Java. Open cmd window and navigate to the folder where HPC client is extracted to. HPC Client can be run in interactive mode or batch mode.

## Interactive Mode:

Go to <HPC\_DME\_APIs>/utils folder and type the following command.

If you are using Unix:

export *HPC\_DM\_UTILS=<Path to HPC\_DM\_UTILS>*

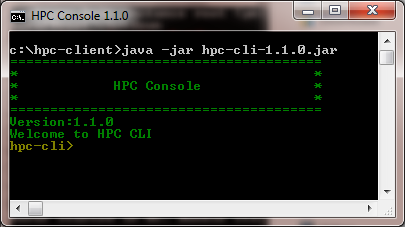
*Java -Dhpc.client.properties=$HPC\_DM\_UTILS/hpcdme.properties -DHPC\_DM\_UTILS=$HPC\_DM\_UTILS -jar hpc-client/hpc-cli-<1>.<1>.<0>.jar*

If you are using Windows:

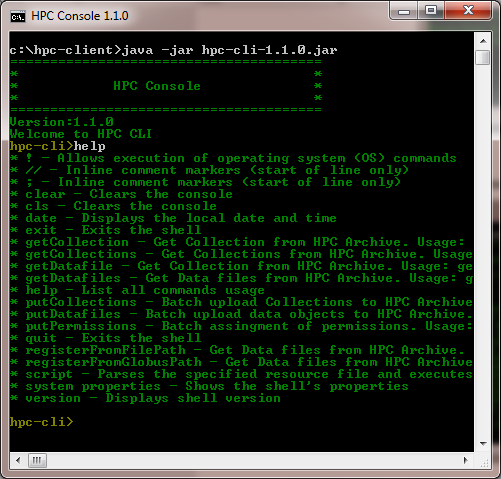
SET *HPC\_DM\_UTILS=<Path to HPC\_DM\_UTILS>*

*Java -Dhpc.client.properties=%HPC\_DM\_UTILS%/hpcdme.properties -DHPC\_DM\_UTILS=%HPC\_DM\_UTILS% -jar hpc-client/hpc-cli-<1>.<1>.<0>.jar*

You will see hpc-cli command line interface as below.



Type “help” to see all the commands supported by hpc-cli.



## Batch Mode

The HPC client supports running multiple HPC Client commands in a batch mode.

*Java -Dhpc.client.properties=$HPC\_DM\_UTILS/hpcdme.properties -DHPC\_DM\_UTILS=$HPC\_DM\_UTILS -jar hpc-client/hpc-cli-1.1.0.jar --cmdfile <Path to HPC commands>*

For example, you may put the following commands into a text file and give that file as --cmdfile input. HPC Client will run these commands in the sequence they are given in the file.

putCollections --source samples/Batch\_Upload\_Input\_File\_Collections.csv

putDatafiles --source samples/Batch\_Upload\_Input\_File\_Object.csv

putPermissions --source samples/Batch\_Permissions.csv

You could also run any operating system commands as part of the file by using “!”. When you are using batch mode, you should set “hpc.login.credentials” property value in hpc.properties. This is to bypass login prompt during batch execution. Please keep credentials file in a safe location so that only you have access to it.

#### Error Codes

HPC CLI returns error code if there is an error executing a given command. Following are the current error codes.

|  |  |
| --- | --- |
| Code | Category |
| CLI\_0 | Authentication error |
| CLI\_1 | Error reading file references from the properties files |
| CLI\_2 | Invalid user input |
| CLI\_3 | No input files to process |
| CLI\_4 | Failed to process collection |
| CLI\_5 | Failed to process data file |

#### Register collections

The HPC client supports registering multiple collections with HPC DME. A collection can be a project, sample, run, dataset or folder. New collection types can be added on HPC DME as needed by a system administrator as described above. The following is the command to register collections.

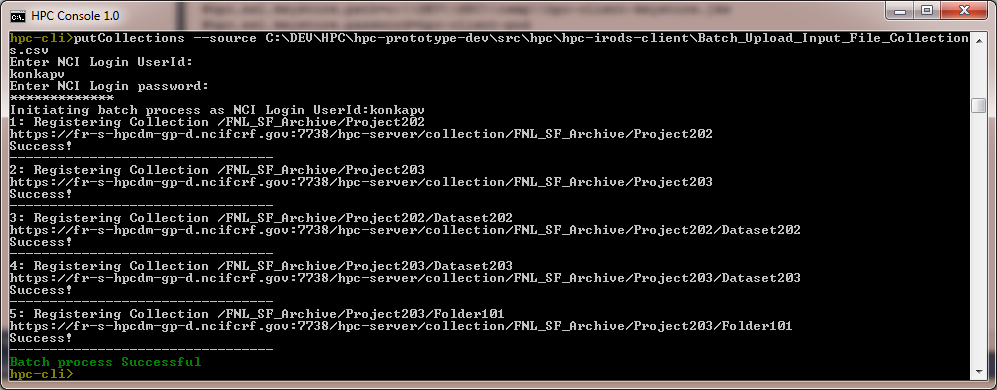
putCollections --source <input file path>

Please refer to <https://github.com/CBIIT/HPC_DME_APIs/tree/master/src/hpc-cli/samples> for sample input files.

Input file for collections registration would need to provide all required metadata along with any user defined metadata. These attributes can be configured on HPC DME as needed by a system administrator.

Collections registration input file can have all collection types in a single file or they can be in separate files. If you are combining all collections into a single input file, attributes not relevant for a particular collection should have empty value.

Running the batch command would give an output as follows.



#### Registering Data Objects

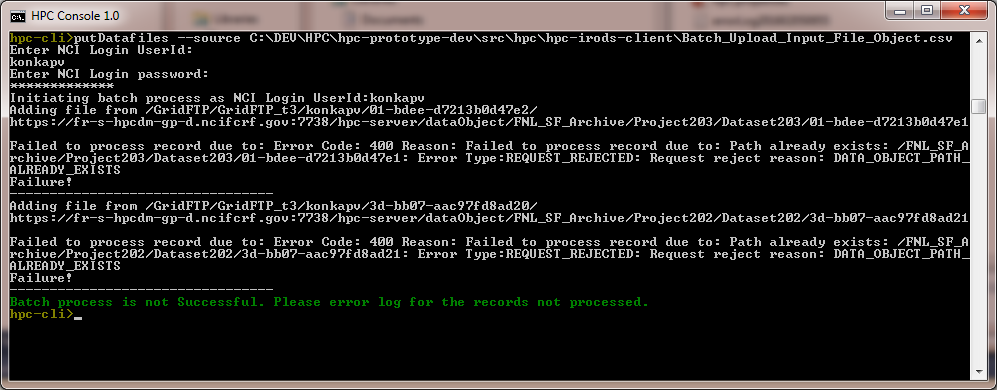
The HPC client supports registering multiple data files with HPC DME. Parent collections needs to be registered first before registering data files. The following is the command to register data files.

putDatafiles --source <input file path>

Input file for data file or folder registration would need to provide all required metadata along with any user defined metadata. These attributes can be configured on HPC DME as needed by a system administrator. HPC DME supports registering data objects from a local file system (synchronously) or from a Globus endpoint (Asynchronously). Based on the given request input, HPC DME client API initiates appropriate request.

Please refer to <https://github.com/CBIIT/HPC_DME_APIs/tree/master/src/hpc-irods-client/samples> for sample input files.

Running the batch command would give an output as follows.



If the source of the data object(s) is a Globus endpoint, the expected operational behavior is that all data objects (files or folders) will be transferred via Globus Connect from the indicated “fileId” (associated with the “fileContainerId”) to the system configured Archive Area.

#### Bulk Registering of Collections and Data Objects from a Globus location

The HPC client supports registering all data files from a Globus location path. This is a convenient command to register all data files from a Globus location. This command recursively goes through each folder and register all files along with folder structure. This process will generate “name” and “modified\_date” metadata along with system generated metadata. This command expects that there is no data hierarchy and metadata validation rules in place for the user DOC.

registerFromGlobusPath --globusEndpoint <Globus Endpoint UUID> --globusPath <Source Path on Globus> --destinationBasePath <Destination Base Path> -- includePatternFile <Pattern file to include source files> --excludePatternFile <Pattern file to exclude source files> --dryRun <true|false>

**globusEndpoint** : Source Globus endpoint

**globusSourcePath**: Source Path on the Globus endpoint. Please make sure to share this path with HPC DME Globus application account.

Production - HPCDME-PROD-APP-ACCOUNT

UAT / DEV - HPCDME-DEV-APP-ACCOUNT

**destinationArchivePath**: Destination path on the HPC DME archive

**dryRun**: Dryrun to see list of source files after processing include and exclude patterns without actually registering the files with the archive. Valid values are “true” or “false”

**patternType**: Include, exclude criteria pattern type. Valid values are Simple, RegEx

**includePatternFile**: List of patterns to match source files from registerFromFilePath to include. In the case of multiple patterns given in the input file, union of all patterns is considered.

**excludePatternFile**: List of patterns to match source files from registerFromFilePath to exclude from included files. In the case of multiple patterns given in the input file, union of all patterns is considered.

**RegEx Pattern:**

With RegEx, the search pattern can be anything from a simple character, a fixed string or a complex expression containing special characters describing the pattern. The pattern defined by the regex may match one or several times or not at all for a given string. Please see the following link for patterns and examples.

https://docs.oracle.com/javase/8/docs/api/java/util/regex/Pattern.html

**Simple Pattern**

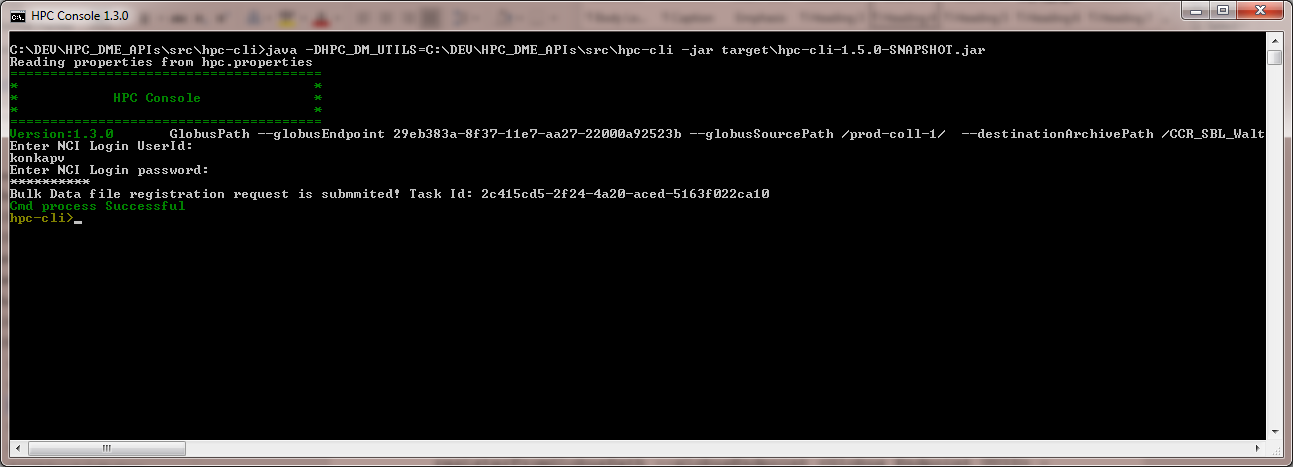
Pattern Matching:

|  |  |
| --- | --- |
| ? | Matches any single character. Eg, "something?" collects any path that is named "something" plus any character. |
| \* | Matches any characters up to the next slash. Eg, "\*/\*/something\*" collects any path that has two directories, then a file or directory that starts with the name "something". |
| \*\* | Matches any characters. Eg, "\*\*/something/\*\*" collects any path that contains a directory named "something". |

Ex:

|  |  |
| --- | --- |
| **\*.bam** | Matches all files with extension “bam” in the source folder |
| **FR00?.bam** | Matches all files with name “FR00” and any one character and with extension “bam” in the source folder |
| **\*\*/\*.bam** | Matches all files with extension “bam” in all folders and sub folders from the source location. |
| **images/\*\*/\*.jpg** | Matches all files with extension “jpg” in images folder and its all sub folders from the source location. |
| **\*\*/sub\*/\*.txt** | Matches all files in sub folder(s) starts with “sub” and with extension “txt” from the source location. |
| **ERI10wef** | Matches with exact file name from the source location |
| **/opt/test/temp.bam** | Matches with exact file name including its path from the source location |

Running the batch command would ask you to enter your NIH credentials to authenticate with HPC DME API. After successful submission, Task Id will be returned.



#### Bulk Registering of Collections and Data Objects from a local file path

The HPC client supports registering all data files from a local file path. This is a convenient command to register all data files or collections from a given path on your file system. This command recursively goes through each folder and register all files along with folder structure following given include and exclude patterns. This process will generate “name” and “modified\_date” metadata along with system generated metadata. This command expects that there is no data hierarchy and metadata validation rules in place for the user DOC. This command registers user provided metadata, if given along with the source files or folders. In order to register metadata, desired file should have corresponding <filename/folder name>.metadata.json file located in the same location as the source file or folder. This command parses the metadata JSON file and registers with along with the physical file or collection. If no metadata is given, this command registers basic metadata – Name, Created with the physical file or folder.

registerFromFilePath -- sourceFilePath<Local file path> --sourceFileList <source File list> -- includePatternFile <Pattern file to include source files> --excludePatternFile <Pattern file to exclude source files> -- filePathBaseName <Base path in the file path> --destinationArchivePath <Destination Base Path> --dryRun <true|false> --confirm <true|false> --metadata<true|false> --threads <number> --archiveType <S3|POSIX> --checksum <true|false>

**sourceFilePath**: Path on the local file system accessible to the CLI.

**sourceFileList:** List of files with full path to register. This file list is relative to **sourceFilePath** value. If you are using sourceFileList**,** do not specifyincludePatternFile or excludePatternFile options

**destinationArchivePath**: Destination path on the HPC DME archive

**includePatternFile**: List of patterns to match source files from registerFromFilePath to include. The pattern can be set to match list of file extensions, like \*.bam. It can also be set to match part of file name. In the case of multiple patterns given in the input file, union of all patterns is considered.

**excludePatternFile**: List of patterns to match source files from registerFromFilePath to exclude from included files. The pattern can be set to match list of file extensions, like \*.bam. It can also be set to match part of file name. In the case of multiple patterns given in the input file, union of all patterns is considered.

Pattern Matching:

|  |  |
| --- | --- |
| ? | Matches any single character. Eg, "something?" collects any path that is named "something" plus any character. |
| \* | Matches any characters up to the next slash. Eg, "\*/\*/something\*" collects any path that has two directories, then a file or directory that starts with the name "something". |
| \*\* | Matches any characters. Eg, "\*\*/something/\*\*" collects any path that contains a directory named "something". |

Ex:

|  |  |
| --- | --- |
| **\*.bam** | Matches all files with extension “bam” in the source folder |
| **FR00?.bam** | Matches all files with name “FR00” and any one character and with extension “bam” in the source folder |
| **\*\*/\*.bam** | Matches all files with extension “bam” in all folders and sub folders from the source location. |
| **images/\*\*/\*.jpg** | Matches all files with extension “jpg” in images folder and its all sub folders from the source location. |
| **\*\*/sub\*/\*.txt** | Matches all files in sub folder(s) starts with “sub” and with extension “txt” from the source location. |
| **ERI10wef** | Matches with exact file name from the source location |
| **/opt/test/temp.bam** | Matches with exact file name including its path from the source location |

**filePathBaseName**: This is the relative base path for your data registration. For example, if your files path is /opt/hpc/folder1/folder2/folder3 and you want your data file relative path to start with /folder1, set filePathBaseName as folder1. The logical file path will be destinationBasePath/folder1/folder2/folder3.

**dryRun**: Dryrun to see list of source files after processing include and exclude patterns without actually registering the files with the archive. Valid values are “true” or “false”

**confirm**: Ask user confirmation to register listing the folders/files. Valid values are “true” or “false”

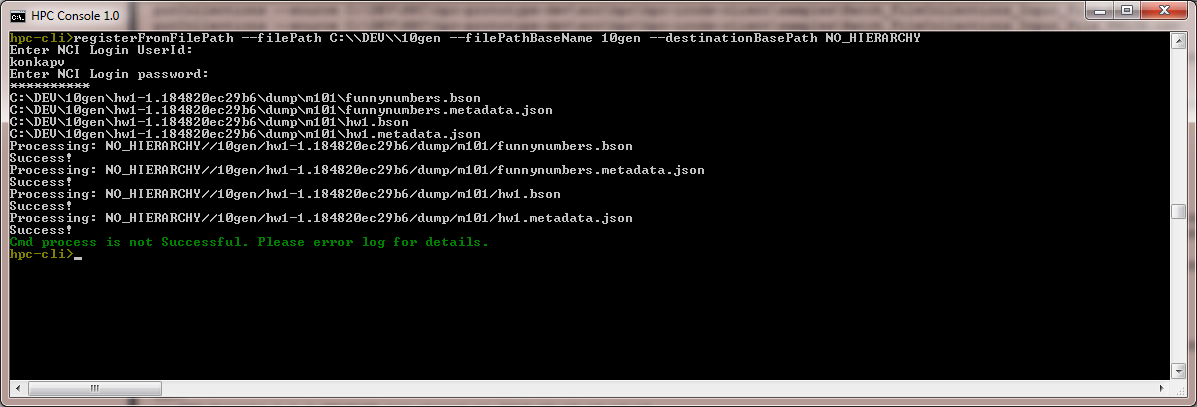
**metadata**: Update metadata flag. If it is set to true, only metadata of the files with matching <file name/folder name>.metadata.json will be updated. No data files will be registered.

**threads**: Number of concurrent threads to run. In the case of multiple source files, each thread will process a file at a time. Make sure to set “upload.buffer.size” value in CLI properties suitable for you JVM settings so that your JVM will not run out of memory while processing large files. By default, this command runs with 1 thread with 1GB heap size on a 4GB memory system.

**archiveType:** Valid values are S3 or POSIX. If you specify “S3”, data is uploaded to S3 type archive using pre-signed URL. If you specify “POSIX”, data is uploaded through API Server and to the final POSIX file system.

**checksum:** By default, data file checksum is computed by the CLI and pass it to the API to verify checksum at the archival time. To turn it off, specify “false”.

Any errors during the process will be written to your HPC\_DME\_UTILS path. Any file that is not registered is written error records log. Running the batch command would ask you to enter your NIH credentials to authenticate with HPC DME API.



#### Update permissions

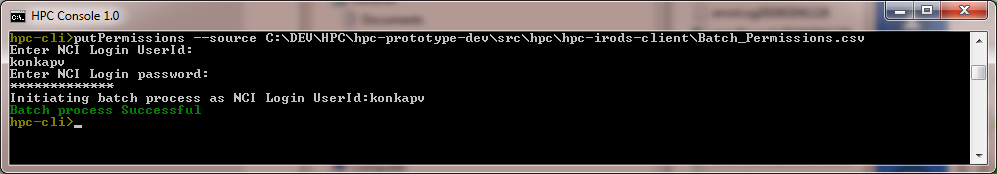
When a user registers a collection or data file with HPC DME, that user assumes “owner” role of the entity. Owner or system admin can assign permissions on these entities to other users or groups. HPC Client provides a convenient command to update permissions on multiple entities in a batch manner. The following is the command to assign permission on collections or data objects to a user or group.

putPermissions --source <input file path>

The following table shows default required attributes for input file. Please see APPENDIX C for sample batch input file.

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| object\_path | Logical path of the entity (Collection or Data file) |
| UserId | HPC DME User Id to assign permission |
| GroupId | HPC DME Group Id to assign permission |
| Permission | Permission value.  Valid values are “write”, “own”, “read”, “none”  Own – Owner of the entity. Can assign permissions to other users  Read – Read only permission  Write – Read and Write permission  None – No permission |

Running the batch command would give an output as follows.



#### Search Collections by metadata

Using the following command, users can search for collections based on a given search criteria and write resulted search records into a file.

getCollections --criteria '<Criteria>' --format <csv|json|txt> --outputfile <file full path> --details <yes|no>

**Criteria**: HPC DME metadata search API criteria is a json string with the combination of search attribute name, search attribute value and the operator to search with. The format of the criteria is as below:

{"a":"<attribute name>","v":"<attribute value>","o":"<Operator>"}

Example: {"a":"description","v":"%prasad%","o":"LIKE"}

Search cmd support searching on multiple criteria conditions. The following is the format:

{"a":"<attribute name>","v":"<attribute value>","o":"<Operator>"}**&&**{"a":"<attribute name>","v":"<attribute value>","o":"<Operator>"}**&&**{"a":"<attribute name>","v":"<attribute value>","o":"<Operator>"}

All the given criteria conditions are applied to the search with AND operator.

**format:** Search cmd takes optional “format” argument to specify format of the output file. Valid options are “csv”, “json” and “txt”. If this argument is not given, output is written in “txt” format. For “csv” format, collections “selfMetadataEntries” are written to the output file. “parentMetadataEntries” will not be written into the output file for “csv” format.

**outputfile**: Search cmd takes optional “outputfile” argument to specify output file full path. If path is not given, default path is “hpc.error-log.dir” + “getcollections\_records”+<Current date> + <type>.

**details**: Search cmd takes optional “details” argument to specify if output should include collection details or not. If not specified, default value is “yes”. If “no” value is given, output file written with collection path values only.

#### Search Collection by path

Using the following command, users can search for collections based on a logical path and write resulted search records into a file.

getCollection --path '<path>' --format <csv|json|txt> --outputfile <file full path>

**path:** Logical path of the collection

**format:** As explained in the section above

**outputfile:** As explained in the section above

#### Search Data objects

Using this command, users can search for data files based on a given search criteria and write resulted search records into a file.

getDatafiles --criteria '<Criteria>' --format <csv|json|txt> --outputfile <file full path> --details <yes|no>

**Criteria**: HPC DME metadata search API criteria is a json string with the combination of search attribute name, search attribute value and the operator to search with. The format of the criteria is as below:

{"a":"<attribute name>","v":"<attribute value>","o":"<Operator>"}

Example:

{"a":"description","v":"%prasad%","o":"LIKE"}

Search cmd support searching on multiple criteria conditions. The following is the format:

{"a":"<attribute name>","v":"<attribute value>","o":"<Operator>"}**&&**{"a":"<attribute name>","v":"<attribute value>","o":"<Operator>"}**&&**{"a":"<attribute name>","v":"<attribute value>","o":"<Operator>"}

All the given criteria conditions are applied to the search with AND operator.

**format:** Search cmd takes optional “format” argument to specify format of the output file. Valid options are “csv”, “json” and “txt”. If this argument is not given, output is written in “txt” format. For “csv” format, collections “selfMetadataEntries” are written to the output file. “parentMetadataEntries” will not be written into the output file for “csv” format.

**outputfile**: Search cmd takes optional “outputfile” argument to specify output file full path. If path is not given, default path is “hpc.error-log.dir” + “getDatafiles\_records”+<Current date> + <type>.

**details**: Search cmd takes optional “details” argument to specify if output should include collection details or not. If not specified, default value is “yes”. If “no” value is given, output file written with data files path values only.

#### Search Data file by path

Using the following command, users can search for data file based on a given logical path and write resulted search record into a file.

getDatafile --path '<path>' --format <csv|json|txt> --outputfile <file full path>

**path:** Logical path of the data file

**format:** As explained in the section above

**outputfile:** As explained in the section above

## Error handling

HPC client interface catches any errors returned by HPC Server API and write them into error log file. The error log is written as <record sequence number> <Error message>.

Those records not processed by HPC client due to any error are written into a separate file “errorRecords<timestamp>.txt” for each command so that only these records can be corrected to go through the HPC batch client again.

# Executing HPC DME API with Command line utilities

Some of the curl command mentioned in the [curl section](file:///C:\Users\luz6\AppData\Local\Temp\HPC_User_Guide-1.docx#_Acessing_HPC_DME) and the batch registration of direcotires mentioned in [HPC DME CLI section](#_Batch_Mode) have been wrapped with bash functions to make them easily accessible from command line.

These functions are available as part of the HPC DM API git repository:

<https://github.com/CBIIT/HPC_DME_APIs/tree/master/utils>

Here are examples on how to use some of the utilities:

## 7.1 One time setup

Follow [thee instructions](#_Pre-Requisites_for_HPC) to get the latest version of the utilities.

Follow the /path/to/HPC\_DME\_APIs/utils/README.md file for the one time setup instructions of the utilities. All utilities start with the prefix “dm\_”.

## 7.2 Generate a Data management API token

Run the following command to generate a DM API token (instead of putting your password every time):

dm\_generate\_token

Note that token expires every two hours.

## 7.3 Generate a Globus token

If you would like to register directories directly from Globus, first you need to edit your Globus information as mentioned in the /path/to/HPC\_DME\_APIs/utils/README.md file. Then run the following command to generate the Globus token:

dm\_globus\_generate\_token

Note that you need to create a Globus account and link it to your NIH account as explained [here](#_using_Globus_for).

## 7.4 Register a collection

To register a collection, use the command:

dm\_register\_collection <collection-description.json> <destination-path>

The git repository contains a sample file for collection-description.json in the directory: /path/to/utils/templates. Feel free to edit the sample file to add more metadata to your collection.

For example, your command can be:

dm\_register\_collection /path/to/HPC\_DME\_APIs/utils/templates/collection-metadata.json /<top-level-directory>/my-collection

## 7.5 Register a dataobject

### Synchronously (From file system)

To register a data object synchronously, use the command:

dm\_register\_dataobject <dataobject-description.json> <destination-path> <source-file>

The git repository contains a sample file for dataobject-description.json in the directory: /path/to/utils/templates.

For example, your command can be (after registering a collection above):

$dm\_register\_dataobject  /path/to/HPC\_DME\_APIs/utils/templates/dataobject-sync-metadata.json /<collection-path>/my-dataobject /path/to/dataobject-file

### Asynchronously (From Globus)

To register a data object asynchronously, use the command:

dm\_register\_dataobject <dataobject-description.json> <destination-path>

The dataobject-descripiton file should contain a valid Globus endpoint and path for your source dataObject. Also, the system account should have a read permission to the Globus endpoint as mentioned [here](file:///C:\Users\luz6\AppData\Local\Temp\HPC_User_Guide-1.docx#_Data_transfer). The repository contains a sample file for asynchronous dataobject-description.json as shown in this example:

dm\_register\_dataobject  /path/to/HPC\_DME\_APIs/utils/templates/dataobject-async-metadata.json /<collection-path>/my-dataobject

## Register a Directory

With the command line utilities you can recursively register all files within a directory and its subdirectories to the archive. By running the next two commands, basic metadata information (e.g., filename, size, etc.) will be created for every file and pushed to the archive. The user is expected to append these metadata when more information is available.

### Synchronously (from file system)

To register a directory from your file system, execute the command:  
  
dm\_register\_directory [OPTIONS] <source-directory-path-on-file-system> <destination-path-on-archive>

OPTIONS:

-d                    Print the files to be included in registration an exit

-i[INCLUDE\_FILE]      Only include the files that match the patterns in INCLUDE\_FILE

 -e[EXCLUDE\_FILE]    Exclude the files that match the patterns specified in EXCLUDE\_FILE

  -h                    Print this message

This command will register all files and directories from the source path to the Archive’s destination path. Regular expressions patterns can be used through the include/exclude options to register only a subset of the files in the source directory. File names that match any of the include regular expression patterns are matched first, then the names that match any of the exclude patterns are removed from the registration job. For more information about the how to write the regular expression patterns, check [this section](#_Bulk_Registering_of).

Metadata can be automatically added for every file that will be registered. The metadata file should have the name <original-file-name>.metadata.json.

The format of the metadata file should be similar to the one defined in the dataObject registration API. Check the HPC\_API\_specification document for more information.

### Asynchronously (from Globus)

To register a directory that exists on a Globus endpoint, first generate a Globus token as shown [here](#_Generate_a_Globus), then execute the command:  
  
dm\_register\_globus\_directory <source-directory-path-globus-endpoint> <destination-path-on-archive> [globus-endpoint]

Note that by default, the Globus endpoint you defined in your hpcdme.properties file is used as mentioned in the README.md. If you would like to use a different Globus endpoint, you can add the optional endpoint UUID at the end of the command.

## Get a collection metadata

To get all the metadata associated with a collection, execute the command:

dm\_get\_collection <collection-path> [response-message] [response-header]

If the [response-message] is omitted, the metadata will be printed to stdout.

## Get a dataobject metadata

To get all the metadata associated with a dataObject, execute the command:

dm\_get\_dataobject <dataobject-path> [response-message] [response-header]

If the [response-message] is omitted, the metadata will be printed to stdout.

## Download a dataobject synchronously

To download a dataObject file from the archive to your current file system, execute the command:

dm\_download\_dataobject\_sync <dataobject-path-on-arhive> <destination-path-on-local-file-system>c

## Download a Collection or dataobject Asynchronously

To download a dataObject or a collection from the archive to a Globus endpoint, execute the command:

dm\_download\_async <source-path-on-arhive> <destination-path-on-globus-endpoint> [globus-endpoint]

Note that by default, the Globus endpoint you defined in your hpcdme.properties file is used as mentioned in the README.md. If you would like to use a different Globus endpoint, you can add the optional endpoint UUID at the end of the command.

# Executing HPC DME API with CURL

Curl is a command-line tool for transferring data using various protocols. It can be used to interact with the HPC DME REST API. If you are using Windows environment, you will need to install tools like CygWin to be able to run Unix-like commands. Please visit <https://www.cygwin.com/> to download. Here are examples on how to use the curl tool to submit REST requests to the HPC DME server.

## Setting up the environment

First make sure you have curl in your environment path. Open a terminal and type:  
$curl

You should get the following line printed on stdout:

curl: try 'curl --help' or 'curl --manual' for more information

If you do not have curl on your path, check your Linux distribution to install it.

## Executing CURL commands

In the examples shown below, any string between the two characters ‘<’ and ‘>’ is a mandatory string that should be replace with an actual value without the enclosing characters ‘<’ and ‘>’.   
  
For example, <server:port> should be replaced with the HPC DME server name and port number (e.g., <https://fr-s-hpcdm-uat-p.ncifcrf.gov:7738)>

In these examples, I dump the returned response header to a file using the -D optional url parameter. For example -D <curl-response-header-file>. Similarly, I dump the response message as a json file using the -o flag. For example: -o <curl-response-message.json>

## Registering a new user

#### curl -H "Content-Type: application/json" -d @<user-info.json> -X PUT <server>/user -H "Accept: application/json" -D <response-header> -o <response-message.json> The input file “user-info.json” should be filled with the new user’s attributes as mentioned in the HPC\_SERVER\_API.

## Registering a collection

#### curl -H "Content-Type: application/json" -d @<attributes-file.json> -X PUT <server>/collection/<collection-path> -H "Accept: application/json" -D <response-header> -o <response-message.json> The “attributes-file.json” should include all the mandatory attributes associated with a collection type.

## Registering a data object

#### As mentioned in the HPC\_SERVER\_API, there are two methods to register data objects: asynchronously using Globus, or synchronously from the file system.

### Registering a data object from a Globus endpoint

#### curl -F dataObjectRegistration=@<attributes-file.json>;type=application/json -X PUT <server>/dataObject/<dataObject-path> -H "Accept: application/json" -D <response-header> -o <response-message.json> Note that the “attributes-file.json” should include the “fileContainerID” and the“fileId” as mentioned in the HPC\_SERVER\_API.

### Registering a data object from the file system

#### curl -F "dataObjectRegistration=@<attributes-file.json>;type=application/json" -F "dataObject=@<dataObject-file>;type=application/octet-stream" -X PUT <server:port>/ /dataObject/<dataObject-path>

#### Note that the attributes-file.json should NOT include the “fileContainerID” nor the “fileId” as mentioned in the HPC\_SERVER\_API.

## Search for collection

Curl -X POST <server>/collection/query/compound –H "Content-Type: application/json" -d @<compound-query.json> -H "Accept: application/json" -D <response-header> -o <response-message.json>

The “compound-query.json” file should contain the attribute to be searched, and the comparison operation as mentioned in the HPC\_SERVER\_API document.

## Search for data Object

Curl -X POST <server>/dataObject/query/compound –H "Content-Type: application/json" -d @<compound-query.json> -H "Accept: application/json" -D <response-header> -o <response-message.json>

The “compound-query.json” file should contain the attribute to be searched, and the comparison operation as mentioned in the HPC\_SERVER\_API document.

## Assign Permissions

#### curl -H "Content-Type: application/json" -d @<permission-file.json> -X POST <server>/acl -H "Accept: application/json" -D <response-header> -o <response-message.json>

## Download data object to a Globus endpoint

#### curl -H "Content-Type: application/json" -d @<destination-description.json> -X POST <server>/dataObject/<object-path> /download --config $HPC\_DM\_TEST/utils/config -H "Accept: application/json" -D <response-header> -o <response-message> The “destination-description.json” file should contain the fileContainerID” and the “fileId” as described in the HPC\_SERVER\_API

## Download data object to a file system

#### curl -H "Content-Type: application/json" -d @<empty.json> -X POST <server>/dataObject/<object-path> /download -H "Accept: application/json" -D <response-header> -o <response-message> In this call, the “empty.json” file should contain just the two characters “{}”. The data object will be downloaded with the name “response-message”

## Delete Data Object (by group administrators only)

#### Adding content

# Executing HPC DME API with Web Client

HPC DME web graphic user interface is developed on the REST APIs to exemplify how end users may utilize the web browsers to fulfil common use case scenarios in scientific data/metadata archiving and data management activities. Please note that what has been presented here is just a reference implementation. Labs, groups or functional units are welcome to implement their own user interface using the REST APIs.

HPC DME UAT Web GUI link for user acceptance testing:

<https://fr-s-hpcdm-uat-p.ncifcrf.gov/>

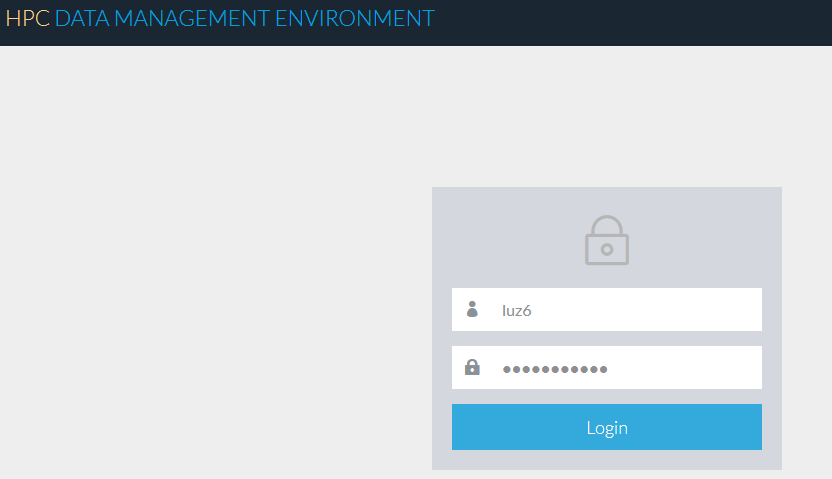
HPC DME Production Web GUI link for production use:

<https://hpcdmeweb.nci.nih.gov>

## Dashboard and login/logout

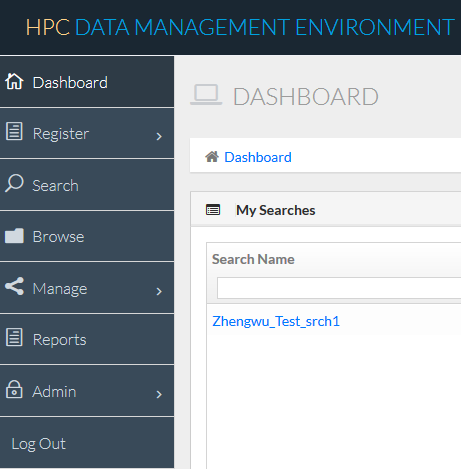
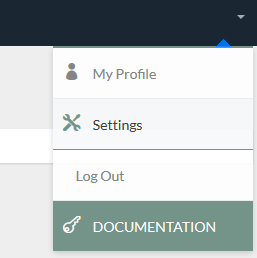
### 9.1.1 Log In

Users are authenticated using the NIH ActiveDirectory – Only NIH AD user IDs are stored at the HPC DME database for tracking authorization. A subsequent iTrust integration will align HPC DME with NIH Secure identity solutions.



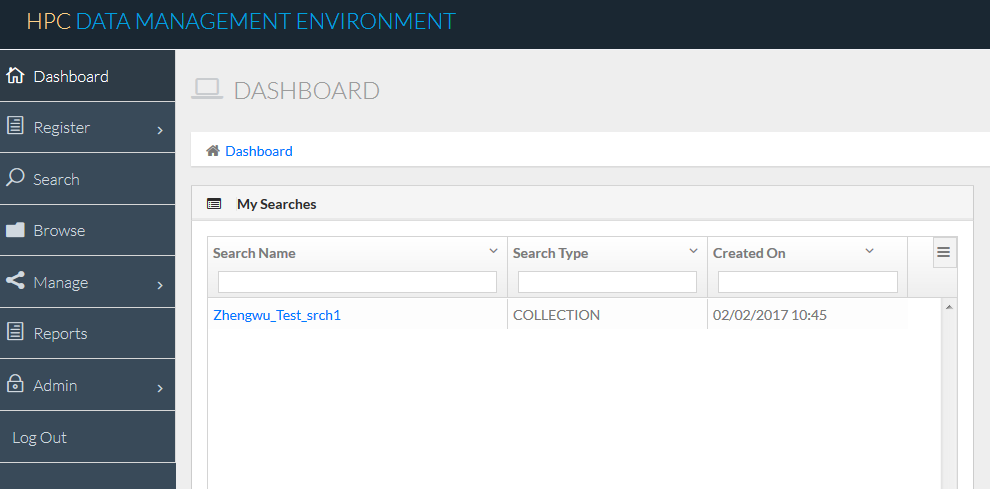
### Log Out

A user may exit out by clicking “Log Out” menu item from left navigational panel or top right corner drop-down menu.

### Dashboard

Default Dashboard shows saved searches and user subscribed events/reports in a listing format for “My Searches” and “My Notifications”, respectively. Clicking any of the listing grid headers will sort the listings in order.



## Search functions

The HPC DME GUI has multiple pages enabling users to enter search criteria and execute targeted searches to identify interested data files/collections. The following summarizes the search capabilities:

1. All text searches are supported case insensitive, i.e., a last name search of “ample”, “AMplE”, or “amPLE” will retrieve any and all users, whose last name in the database is “ample”
2. Wild card “\_” represents any one character and “%” indicates 0, 1 or more characters
3. For group administration only, all records in the domain User or Group tables will be listed if no search criteria are entered for any of the searchable text fields

HPC DME API/GUI client supports the following functions:

### Google Style Search

Users may leave the default entry as “Any” for the “Attribute Name” dropdown, but enter any text at the “Attribute Vale” box for a google style search, either for Collection or Data Object search type. Note: S/he needs to select an operator from the “Operator” dropdown.

### Basic Search

Users may perform a typical [hierarchic](https://www.google.com/search?q=hierarchical&spell=1&sa=X&ved=0ahUKEwif36GIgNPTAhXLQyYKHYdzApgQvwUIJCgA&biw=1280&bih=560) search by entering a level attribute per their DOC organization of collections structure. This will help drill down the list of “Attribute Name” by showing only corresponding attribute names for the level selected earlier. A User may add additional search filter criteria via clicking the button”Add Criteria” and entering his/her search “Level”, “Attribute Name”, “Operator” and “Attribute Value”. Multiple search parameters will be combined with “and” to further nail down the list of search results for either search type “Collection” or “Data Object”.

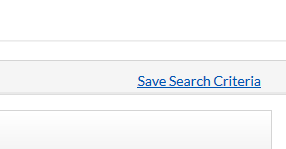
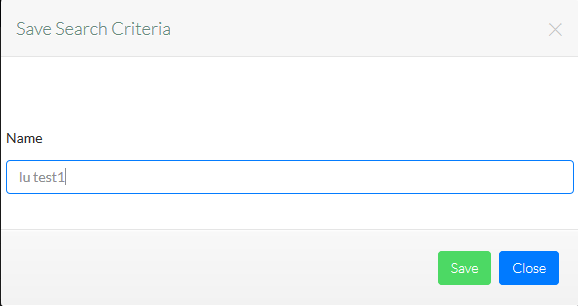
### Advanced Search

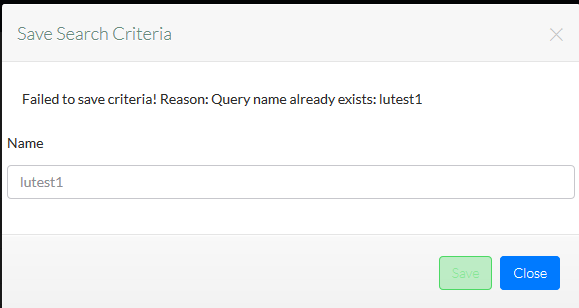
Users may perform an advanced search by typing the previously entered search ID at the basic search section such as “A1”, “A2”, “A3” but entering his/her own desired filtering logic operator “AND”, “OR”. Users may also use parentheses to indicate the filtering order/sequence. For instance, the expression “(A1 OR A2) and (A3 or A4)“ will return any search results meeting both criteria “(A1 or A2)” and “(A3 or A4)”, where the filtering inside the parentheses will be assessed first.

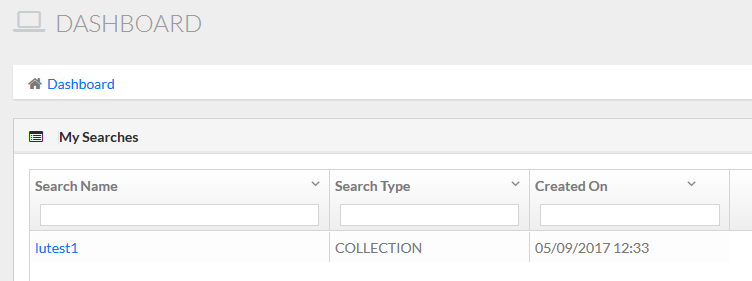
### Saved Search

Users may save an executed search by clicking the hyperlink “Save Search Criteria” and then typing a name for the search for late retrieval. Users may get a prompt message indicate “Query name already exists” if they are creating the same search name. Any saved searches will be displayed at the home page “Dashboard My Searches, where the Search Name, Search Type and Created On (Date/Time) stamps are listed for information purpose.

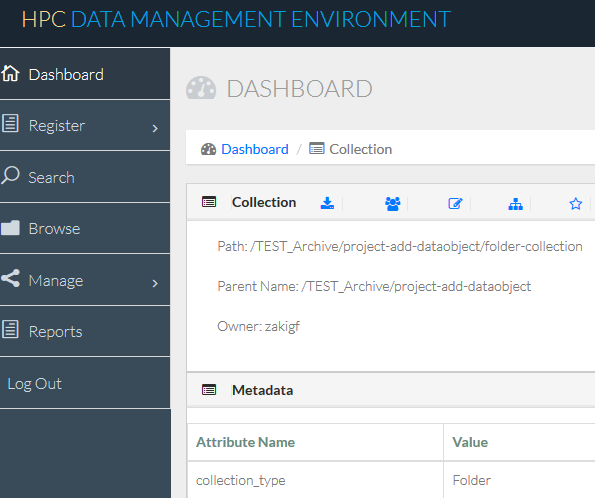
The Save Search feature may be enhanced further to include capability of editing(deleting) and sharing among users.

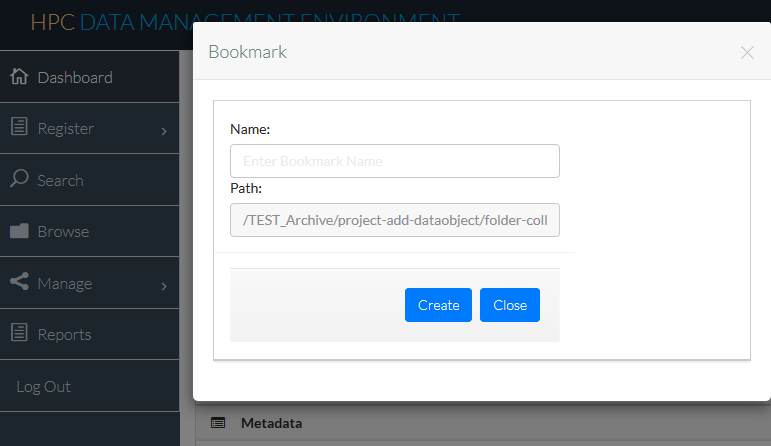
 

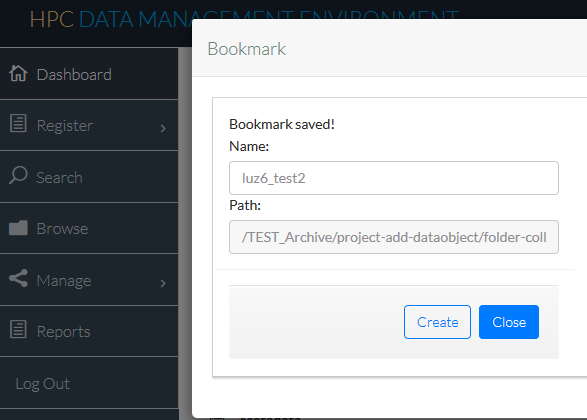




### Bookmark Function

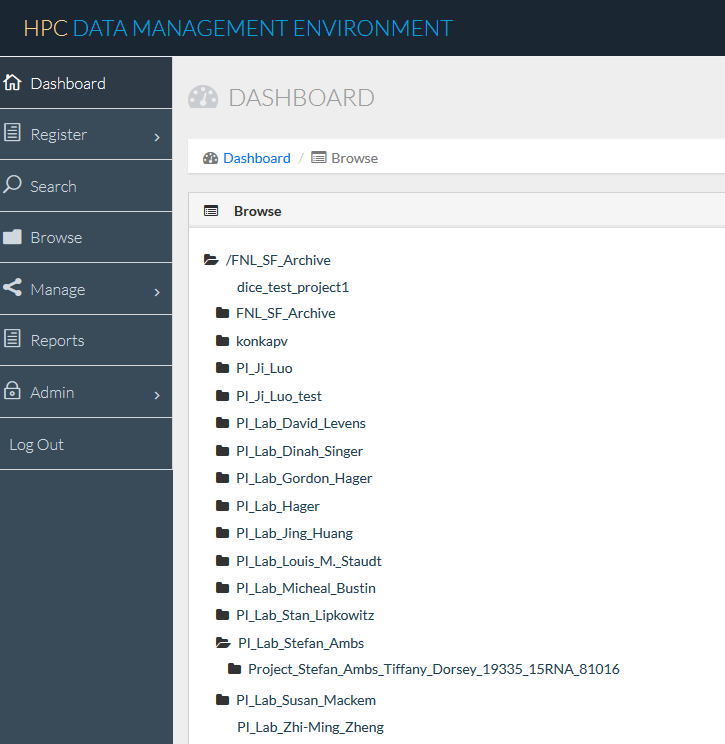
Users may bookmark any type of search results and enter a bookmark name for easy later retrieval or use the browse feature to drill down the specific branch tree per their own authorizations designated. To perform such function, a user needs to select the “Star” icon at the details page of a collection or data file. S/he enters a name for his/her collection or data file path, and then enter Create to have the bookmark saved and tracked in the database. Note: Bookmark name is not allowed to have empty space between words but may be concatenated with “\_” or “-“



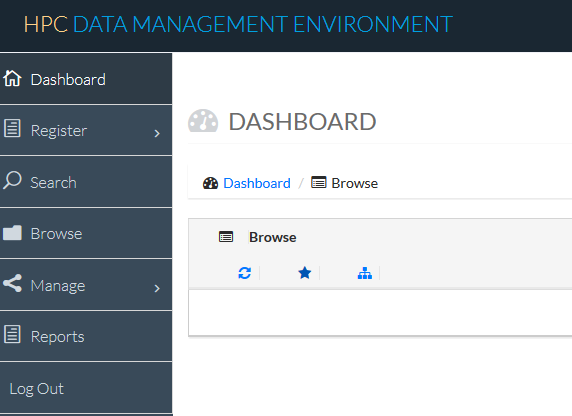


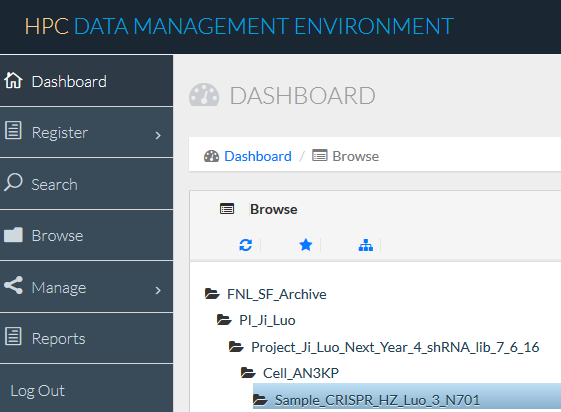
## BROWSE functions

The Browse menu offers a user to navigate their DOC data/metadata hierarchies through a tree-branch expandable user interface and allows users to drill down through the data organizational layers unique to their own DOC policy.

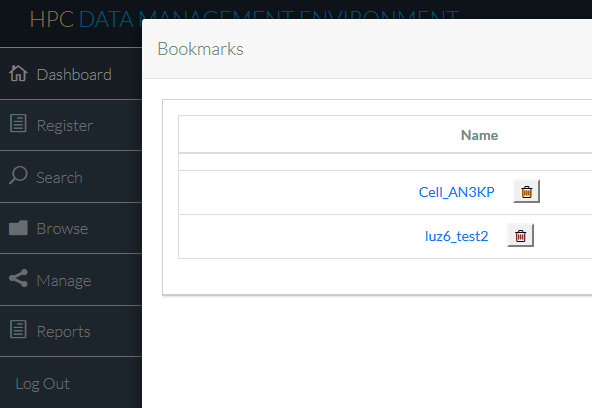


If a user does not have a root access to his/her own DOC or other groups’ base path but wishes to browse through navigating children collections (directories) s/he has access to, s/he may use the Bookmark function to bookmark the collections URL first via Search menu, select the Bookmark icon in the Browse menu as indicated below, identify the bookmark to drill into from the listing table, and then click the Name for the intended bookmark.



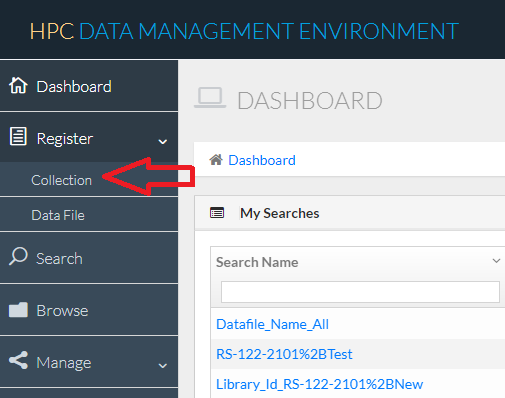


Please note that saved bookmarks may be deleted from the Browse menu if they are no longer required by clicking the “Delete”icon at the right side of the bookmark name as indicated below..

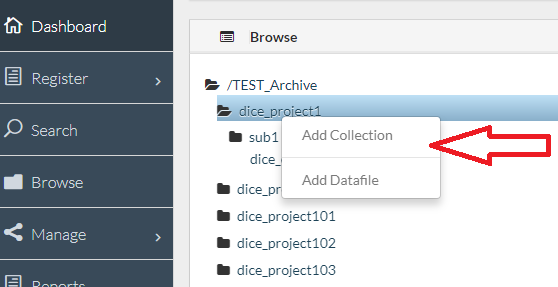


## Register a collection

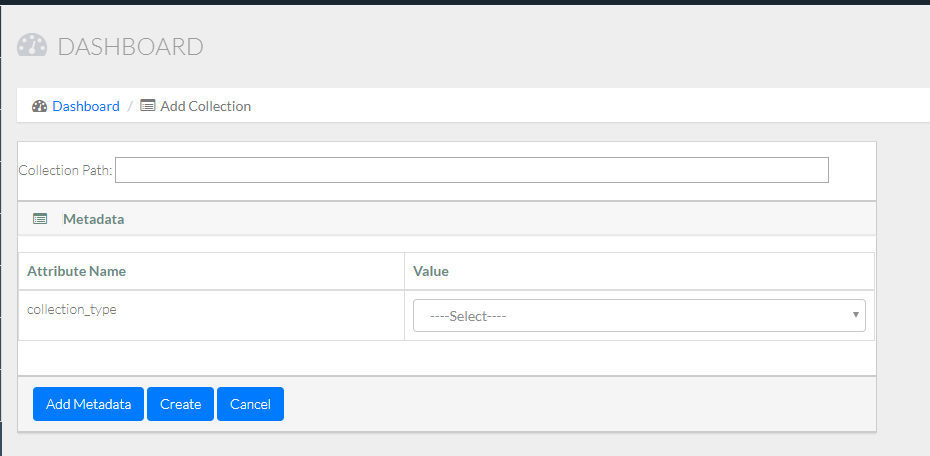
Users may create a collection through Web UI. A collection can be created by clicking on “Collection” link under “Register” side menu option. You are responsible for typing valid logical collection path in the “Collection Path” text box. Preferably, you may want to browse and copy the path info from an existing collection from the top section after “Path:” (including the “/” sign). By default, “collection\_type” list box is populated with valid collection types defined in your metadata policy file. If no collection types are defined, “Folder” collection type will be displayed.



You may also create a collection by clicking on “Add Collection” option in the pop-up menu item when you right mouse click on an existing collection on the Browse page .



If you are creating a collection through right mouse click option, you are responsible for appending valid collection name to the logical collection path populated in the “Collection Path” text box. The “collection\_type” list box is populated based on the parent collection valid sub collection types defined in your metadata policy file. If no sub collection is allowed to the selected parent, the “Add Collection” page displays an error message.



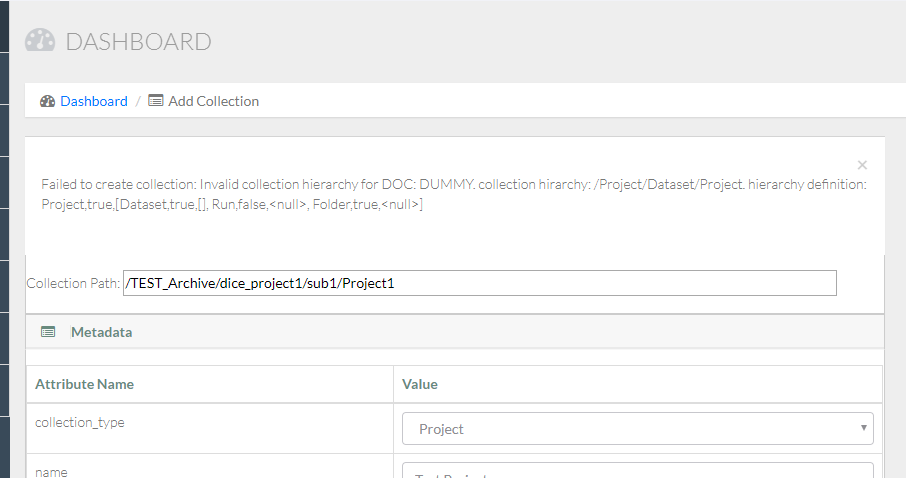
UI form is refreshed when you select a collection type. The form displays all required metadata attributes based on your DOC policy file. Any metadata attribute with “default value” is populated with the value in its text box when displayed in the form. Valid values defined for a metadata attribute are displayed as list box in the form. “Add Metadata” button may be used to add user defined metadata attributes to the collection. Duplicates attributes names are not allowed. Clicking on “Create” button validates the form and submits it to the Server API and displays its response. If there is an error from the Server, the form retains its state while it displays the error message.

The “collection\_type” attribute is required and its value is populated based on DOC hierarchy defined in its policy file. If no hierarchy is defined, “Folder” is the default “collection\_type”. When creating a collection, if you did not select a valid “collection\_type”, the registration will result in an error similar to the following:

Failed to create collection: Invalid collection hierarchy for DOC: DUMMY. collection hirarchy: /Project/Dataset/Project. hierarchy definition: Project, true, [Dataset, true, <null>, Run, false, <null>, Folder, true,<null>]

First highlighted message shows the hierarchic structure you are trying to create and the second highlighted message shows the valid hierarchy. Valid hierarchy is displayed in String representation of JSON object. The required format is [“collectionType”, “isDataObjectContainer”, subCollectionsHierarchies]. For the above example, it indicates that:

1. Project is the parent collection under the DOC’s base path (defined by the DOC policy)
2. The Project collection may have data files deposited under.
3. The Project collection has 3 separate subcollections, i.e., “Dataset”, “Run”, and “Folder”.
4. The Dataset subcollection may have data files deposited under but no children collections under.
5. The Run subcollection will not have data files deposited under . Nor will it have children collection under.
6. The Folder suncolection may have data files deposited under but no children collections under.

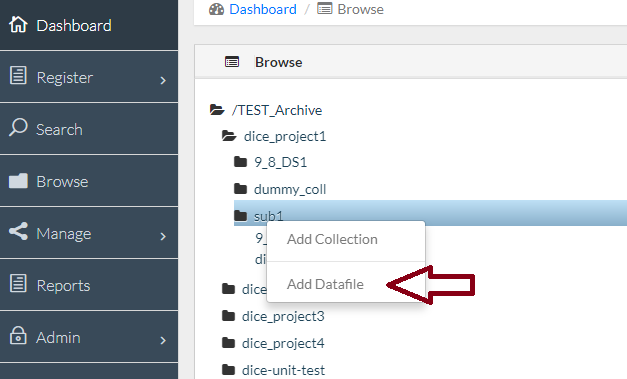


## Register a Data File

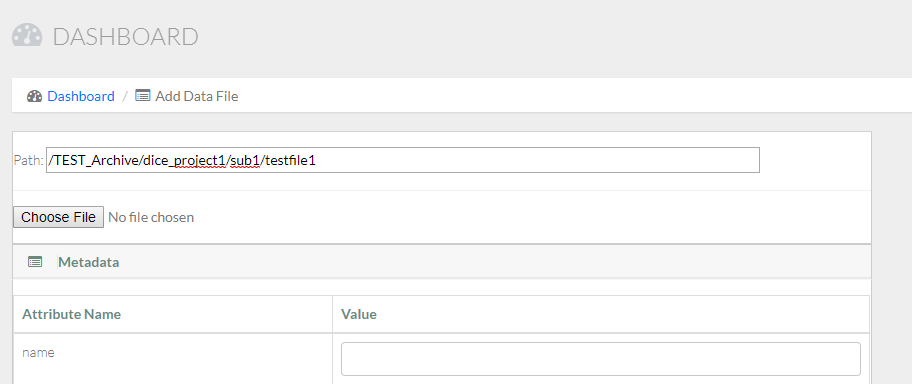
Users may register a data file through Web UI. A data file can be created by clicking on “Data File” link under “Register” side menu option. You are responsible for typing valid logical data file path in the “Path” text box. Preferably, you may want to browse and copy the path info from an existing collection from the top section after “Path:” (including the “/” sign).



You may also create a data file by clicking on “Add Datafile” option in the pop-up menu item when you right click on an existing collection on the Browse page to add the data file to.



If you are creating a data file through browse option, you are responsible for appending valid collection name to the logical collection path populated in the “Path” text box.



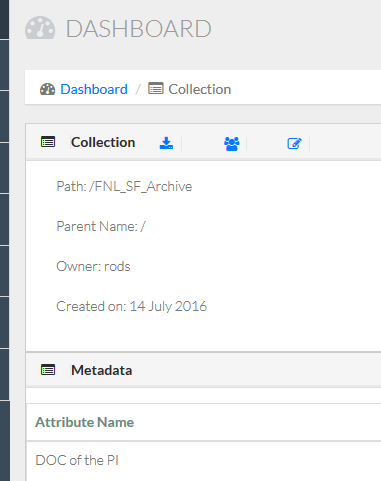
With this registration page, you can only register one file at a time. You will have to click on “Choose File” button to select a file from your local machine. The registration form displays all required metadata attributes for a data file based on your DOC policy file. Any metadata attribute with “default value” is populated with the value in its text box when displayed in the form. Valid values defined for a metadata attribute are displayed as list box in the form. “Add Metadata” button may be used to add user defined metadata attributes to the data file. Duplicate attributes names are not allowed. Clicking on “Create” button validates the form and submits it to the Server API and displays its response. The form is submitted to the server along with the data file selected. This data file is uploaded to the server synchronously. If you are uploading large files (more than 100MB), it is recommended to use asynchronous registration option through API for faster response.

If there is an error from the Server, the form retains its state while it displays the error message. You may correct the error and submit the form again.

## SHARING functions

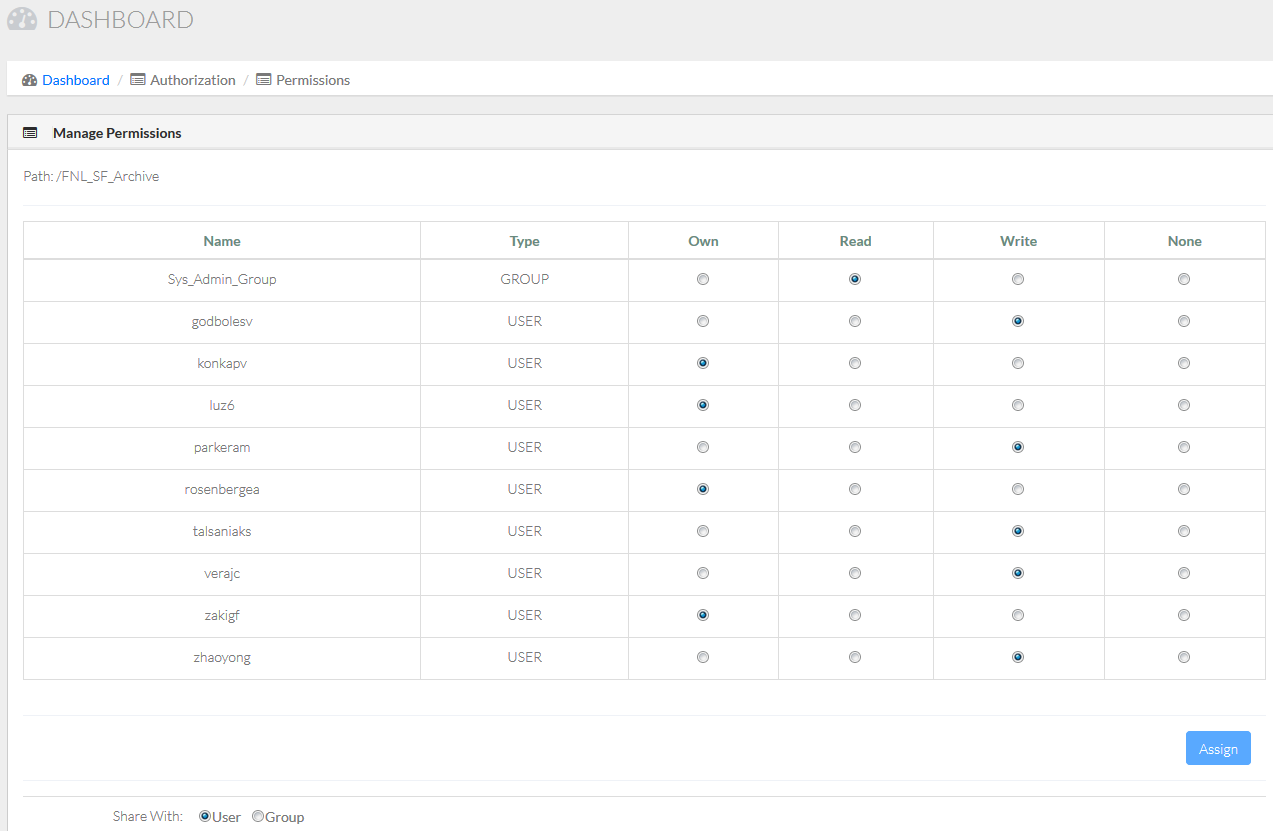
The sharing function is a critical part of the HPC scientific data management and collaboration at NCI. The emphasis and workflow relies on the fact that each data owner/producer wants to decide how they share their data. Check out the HPC DME Web GUI -Download, Sharing and Update Metadata icons at below snapshot.

***HPC DME Web GUI -Download, Sharing and Update Metadata icons***



A data owner may assign who (either via an individual user account or group membership) can own, write, read or NONE certain collections or data files through the GUI Browse or Search menu items. The middle icon in the above figure along the “Collection” indicates that the login user may be authorized to assign access or to have read permissions on the path specified. If a user has read permissions only, s/he will not be able to select/unselect corresponding radios of permissions.

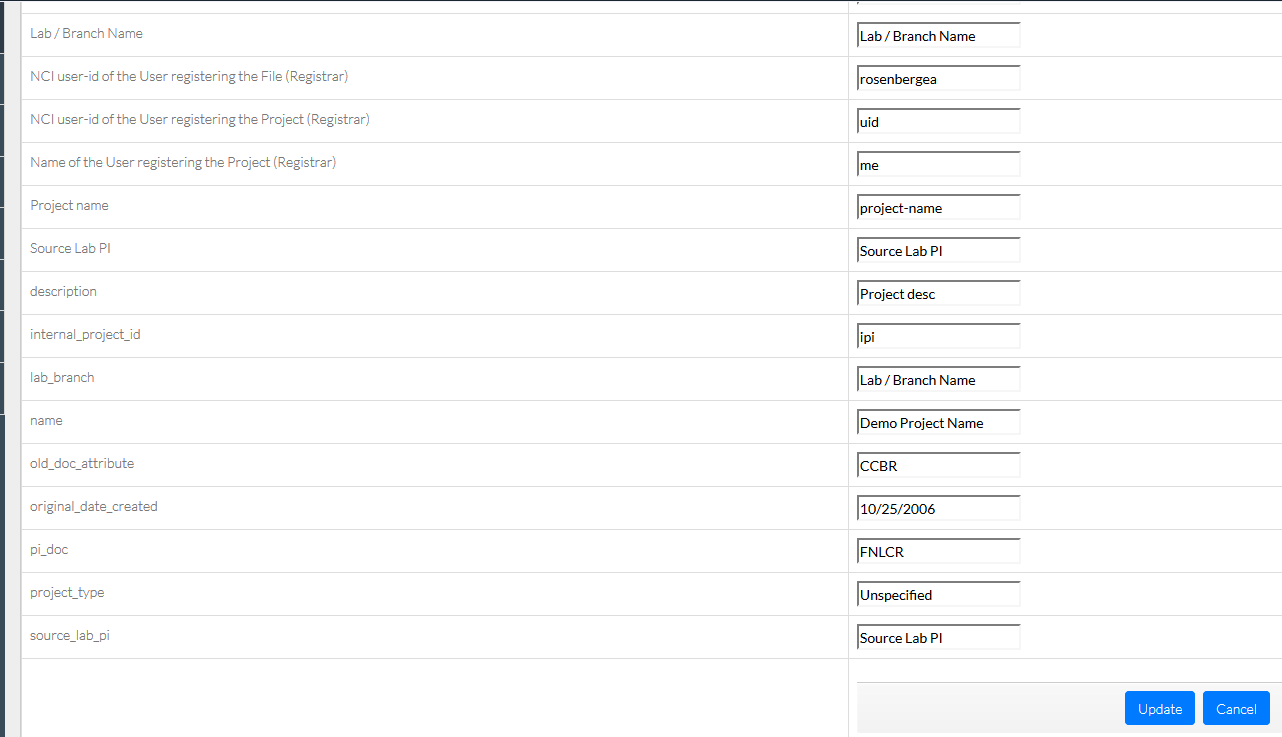
Clicking that icon takes the user into the below “Manage Permissions” figure of showing all the existing users and/or groups with either of the following privileges: Own, Read, Write, and None. From here, the user may assign a user or group either of the 4 permissions to the indicated collection path or data file (object).



## Update metadata in non-batch mode

The web GUI allows any users, through either individual user account’s authorization or group membership, to be able to make update to metadata attribute values. The right icon at previous figure HPC DME Web GUI -Download, Sharing and Update Metadata icons along the “Collection” indicates that the login user is authorized to update custom metadata attributes on the path specified.

Clicking that icon takes the user into the below “Update Metadata” page showing all enabled text box entries of existing custom metadata attributes for value update. From here, the user may enter any new values for their interested metadata attributes and click the “Update” button at bottom right to commit the transaction/update on the backend database. Please note that certain system metadata attributes such as CheckSum is for display purpose only Nobody will have permission to perform update on these system generated metadata attributes.



## Download Collection/file from archive

## Share a Globus End Point with HPCDME Application Account

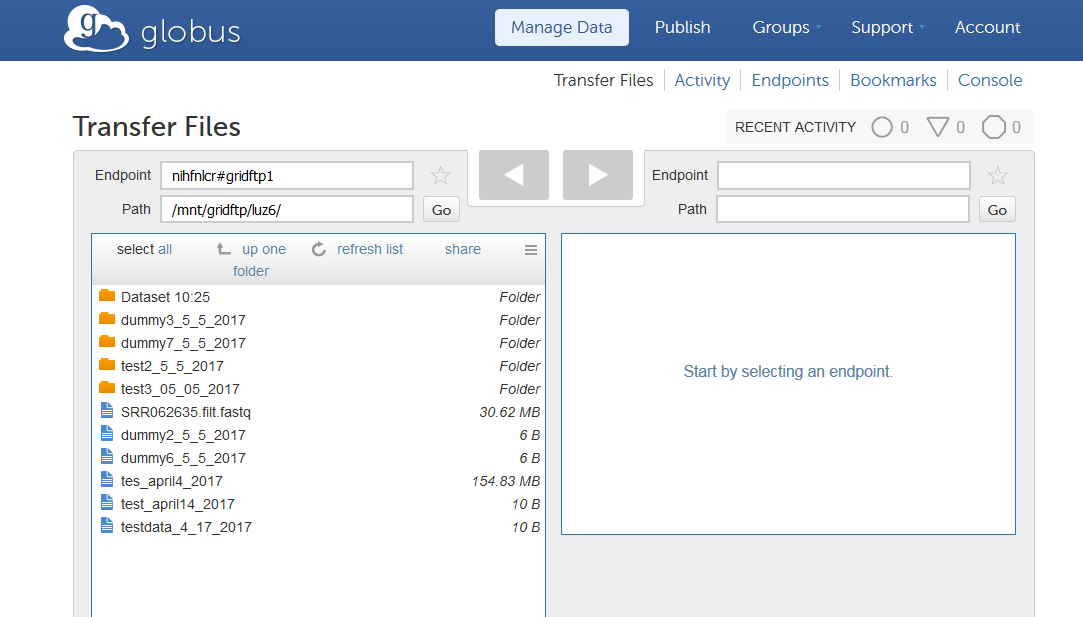
When a user uploads a large data file into archive from an existing Globus end point, the actual data transfer may be performed asynchronously via s system application account. The user needs to grant the application account read access to their source end point before the upload request is submitted. Similarly, a user needs to grant the application account write access to their target Globus end point before a collection or file download request is submitted. When a user downloads a large data file from an Archive into the target Globus end point, the actual data transfer may be performed either synchronously or asynchronously via Globus’ system application account.

***HPC DME Globus Application Accounts Summary Information:***

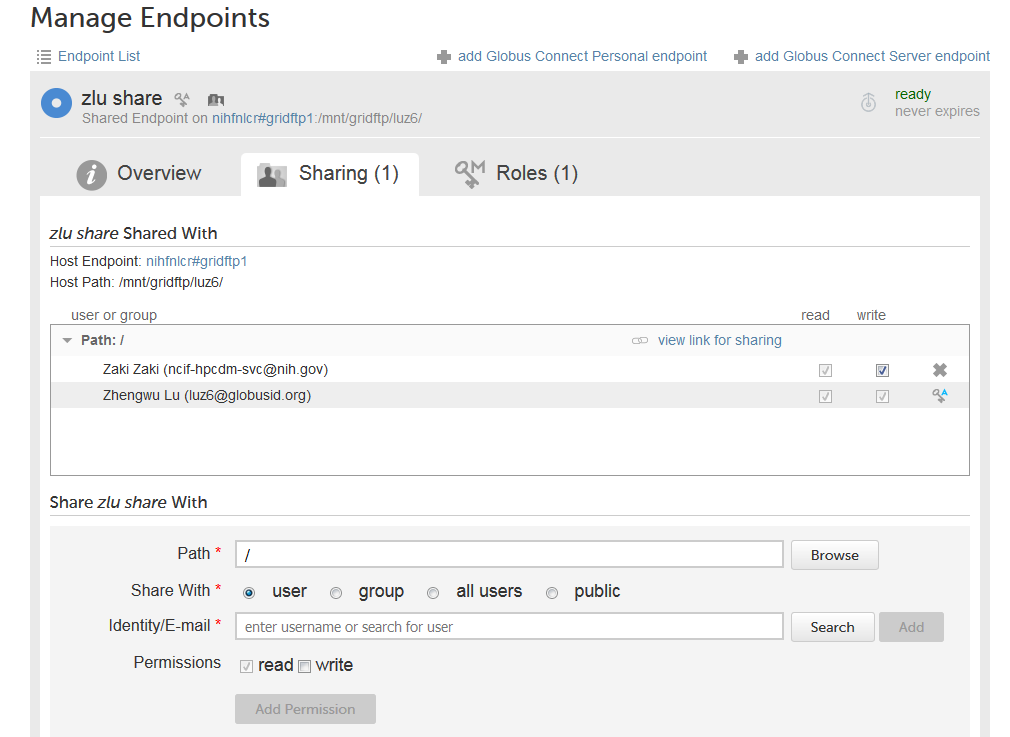
|  |  |  |
| --- | --- | --- |
| Globus Application Account | Globus Application ID String | Applicable HPC DME Tier (s) |
| HPCDME-DEV-APP-ACCOUNT | 025ff462-07e1-483b-8dbb-1fc26c7eb17e@clients.auth.globus.org | Development/UAT/Training |
| HPCDME-PROD-APP-ACCOUNT | 07cec4f4-8d99-4b9d-a513-d9a1148ac054@clients.auth.globus.org | Production only |

The following summarizes the key procedures for a user to share a file or collection(folder/directory) with the HPCDME application account:

1. Create a Globus share of an existing directory, where you want to share with the Application Account through Globus “Transfer File” menu.



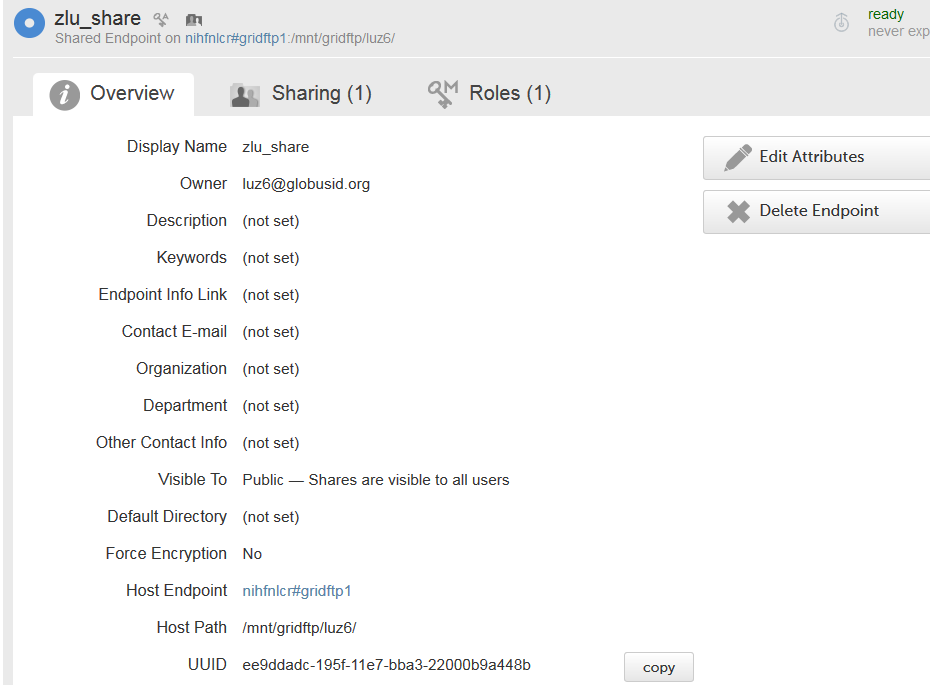
1. Share with HPC DME application account through Globus “Manage Endpoints” Sharing tab following the creation of the share.

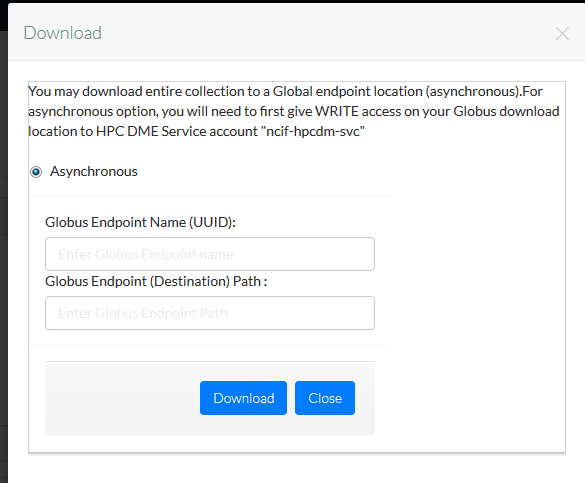


1. For “Path”, you may leave as is to share the original directory or enter/browse to a subdirectory. Unfortunately, Globus does not allow sharing a file or files directly.
2. For “Share With”, you want to check the radio button “user”
3. For “Identity/E-mail”, you want to enter or search the corresponding HPC DME application account email depending on whether you are running the solution at UAT or Production
4. For “Permissions”, you will select the default “read” or check the “write” permission: Uploading a file/collection to an archive requires granting read access to the source end point (s) while downloading from Archive requires granting write access to the application account on the target end point (s).

## Download a Collection of Data Files

The web GUI allows any users, through either individual user account’s authorization or group membership, to be able to download a collection of files (via Globus asynchronous transfer) or a single file (via either synchronous or asynchronous transfer). When performing a Globus transfer, a user may copy the UUID from the target endpoint Globus “Overview” tab UUID field and paste that as the input value for “Globus Endpoint Name (UUID)” at the Download dialog. S/he needs to manually enter the entry for “Globus Endpoint (Destination) Path” at the prompt dialog. The collection of files will be transferred to the target Globus endpoint with a directory name created using what is entered. The left icon at Figure HPC DME Web GUI -Download, Sharing and Update Metadata icons along the “Collection” indicates that the login user is authorized to download the selected collection (of data files) to a Globus target end point.

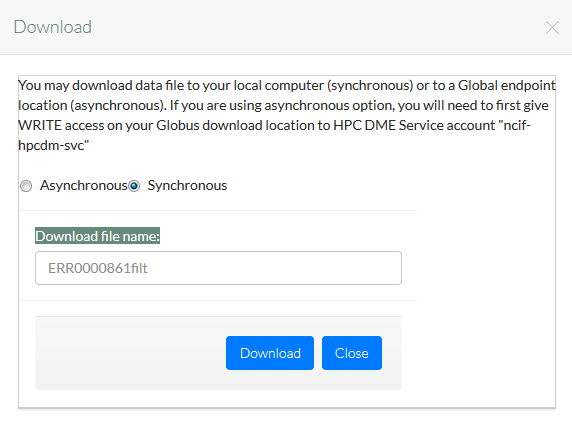


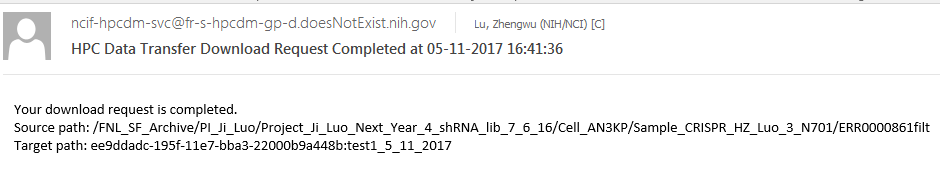


## Download a Single Data File

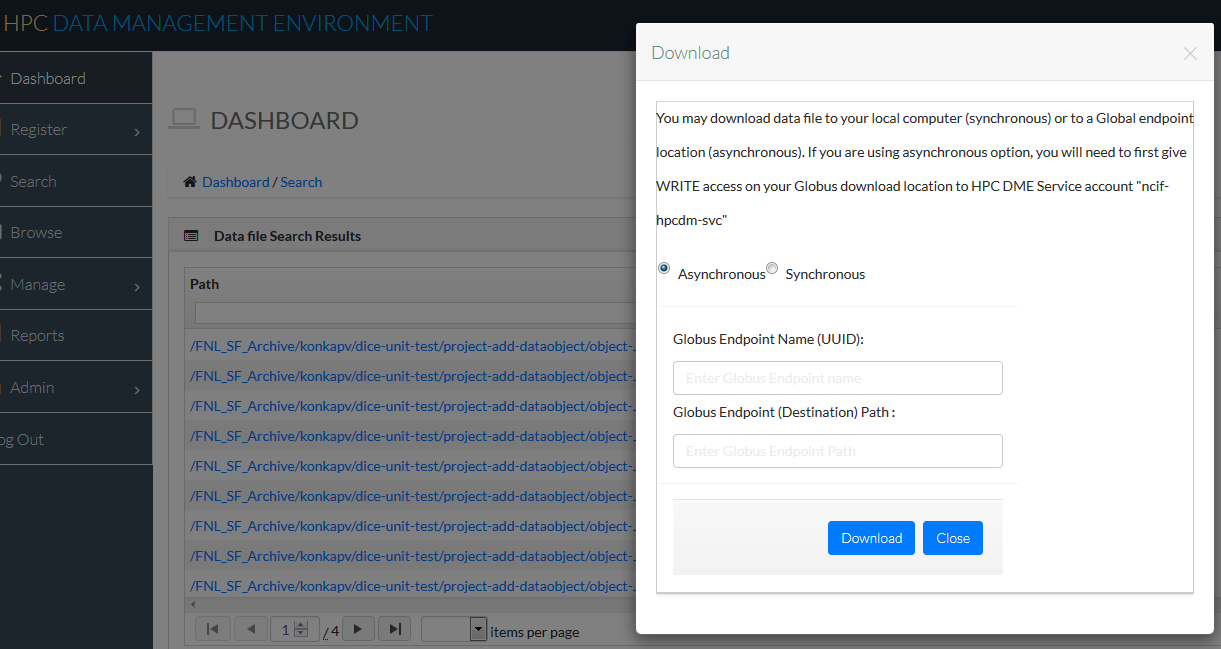
To download a single file, you want to navigate through the HPC DME Search menu and check Search Type “Data Object”. After executing a search, you click the “Download” icon on the adjacent file to initiate the download action. Subsequently, you decide how to download the data file: Synchronous download to local share or asynchronous to a Global endpoint location.

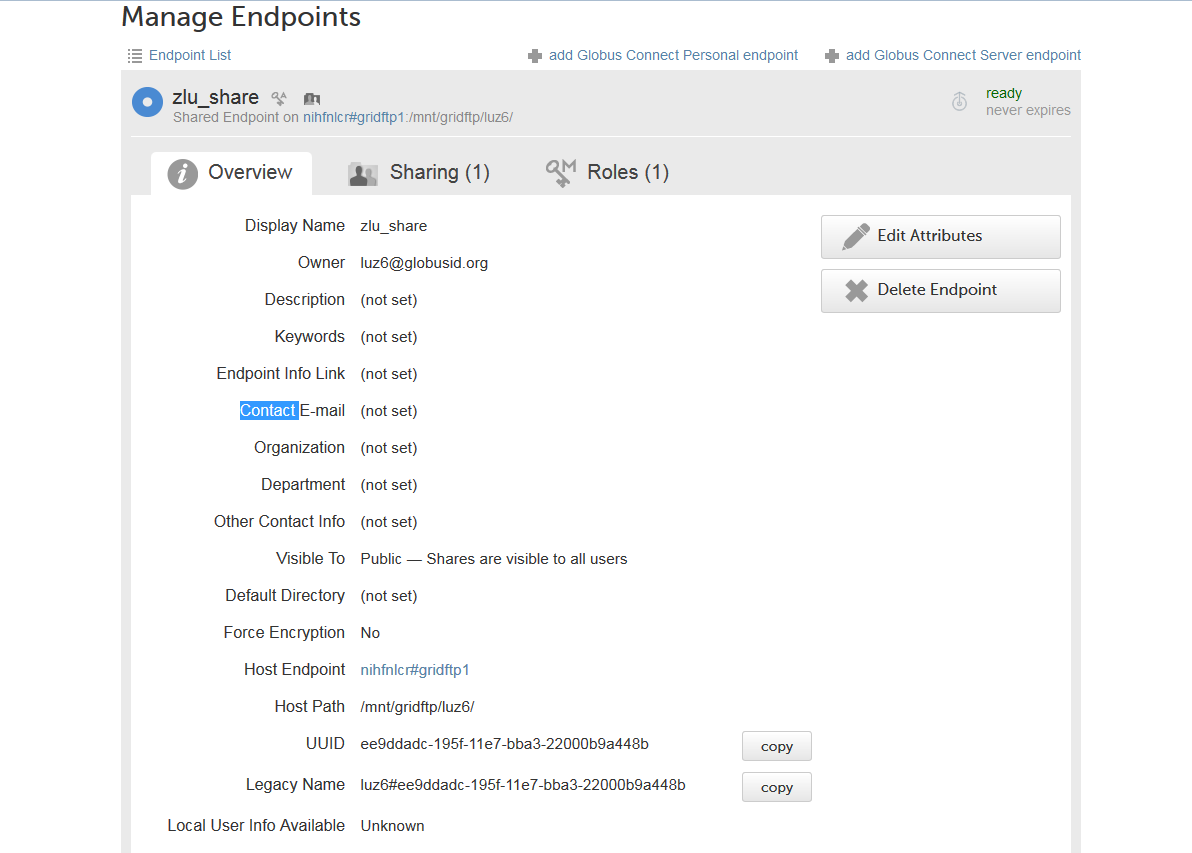
For synchronous download(s), a user has the option of adopting the existing file name or typing over to use a different name. Once the file name is entered, clicking download will transfer the data object from Archive into your local drive/share using the determined file name. If s/he has subscribed to the event type “Data Transfer Download completed”, the user will be informed by email after the download is either completed or runs into failure with relevant contextual information of the collection downloaded.





If a user uses asynchronous option, s/he will need to first give WRITE access on their Globus download location to HPC DME application account beforehand. The user may copy the UUID from the target endpoint Globus “Overview” tab UUID field and paste that as the input value for “Globus Endpoint Name (UUID)” at the Download dialog. S/he needs to manually enter the entry for “Globus Endpoint (Destination) Path” at the prompt dialog. The single file will be transferred to the target Globus endpoint with the file name created using what is entered. If s/he has subscribed to the event type “Data Transfer Download completed”, the user will be informed by email after the download is either completed or runs into failure with relevant contextual information of the file downloaded.

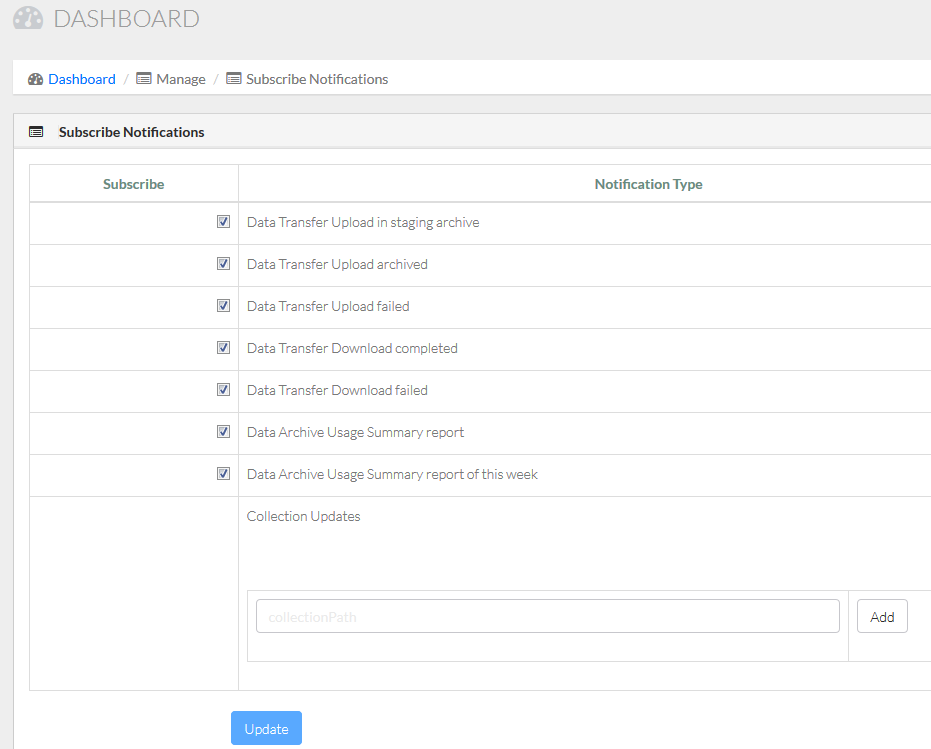
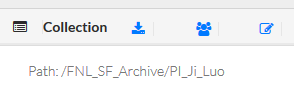




## Notifications and reports

### Notifications Management

A user manages his/her own notifications by subscribing/unsubscribing to the different event types related to data transfer (upload/download), reporting, or data/metadata updating occurring on their interested collection (s). If a user subscribes to a parent collection for notification, any updates occurring on the parent and all its children collections will be informed due to the inheritance feature implemented. To subscribe/unsubscribe to an event type, a user needs to check/uncheck the adjacent checkbox. To subscribe to a collection update of your interest, you want to browse, navigate to the collection details page, copy what is after the “Path:” into the clipboard, and then paste this path into the “collectionPath” text box. You finish the transaction by clicking Add and then Update buttons on the “Subscribe Notifications” page. To unsubscribe, you want to uncheck those collection paths no longer interested, then click Update to commit

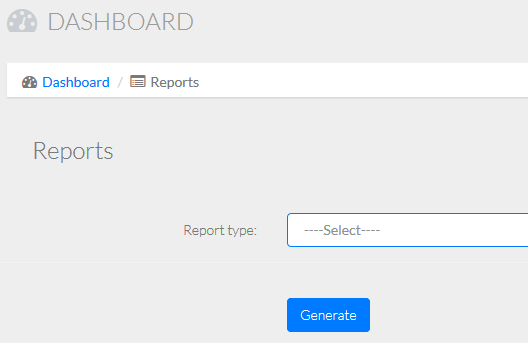
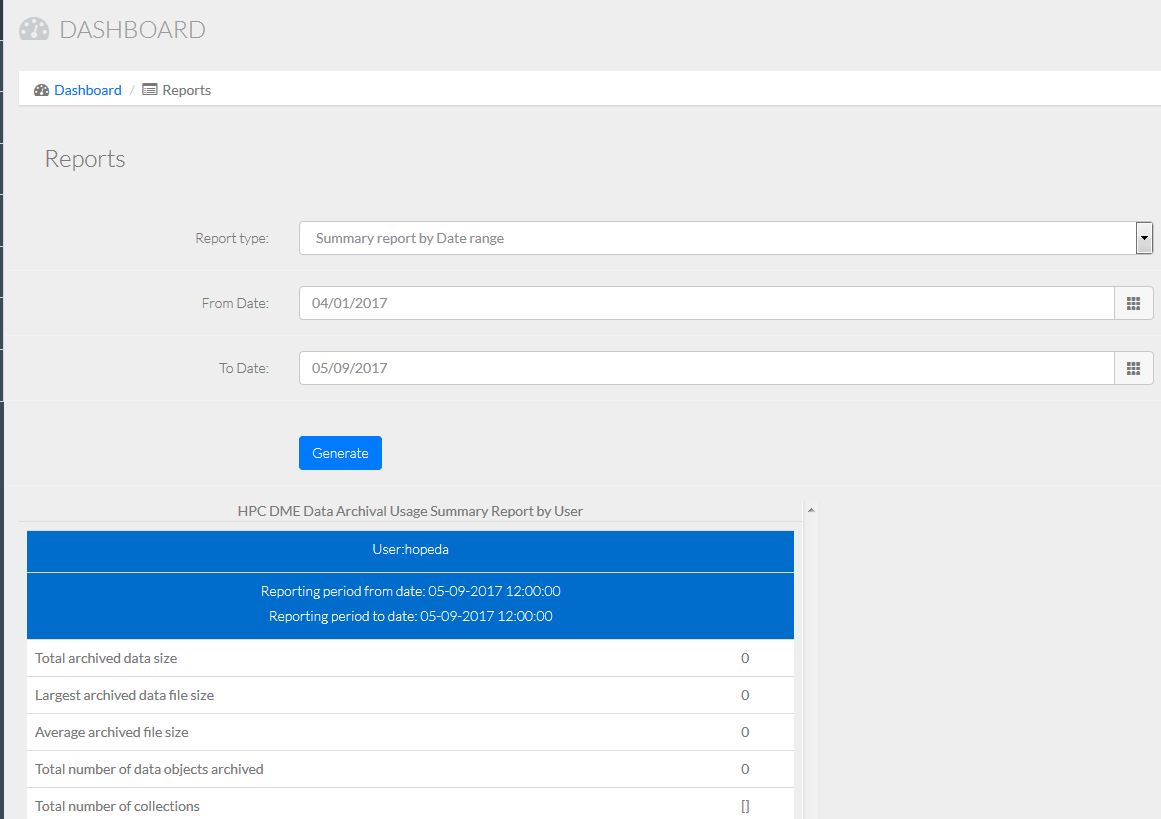
 

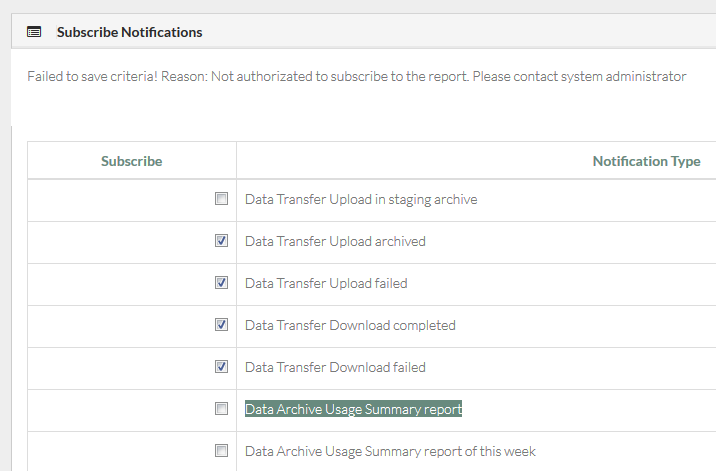
### Reports Management

A user may access existing reports through Report menu and pick a report from the following Report types: Summary Report; Summary Report by Date range; User Report; and User Report by Date range; Base path report; and Base path report by Data range; Collection path report; and Collection path report by Date range. A report dialog will open up prompting for a user selection or entry if a User Report is initiated or for a date range entry if either Data range report is in action (Summary or User Report). You present the result by clicking the Generate button on the report dialog page. Additional reports may be requested and custom developed/deployed in future.

Note: Two reports “Data Archive Usage Summary report” and “Data Archive Usage Summary report of this week” are developed for system administrators only, for which non-system administrator users will get error prompt if they try to subscribe to them.

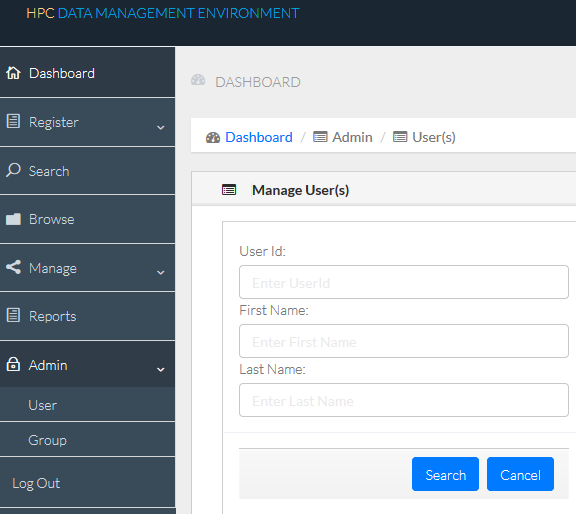
“Total registered Users” value in the report generated by Base path represents number of users assigned to the selected Base path as default base path. “Total registered Users” value in the report generated by Collection path represents number of users registered collections or data files with in the given collection path.



## Group administrator functions

The web GUI allows a group administrator to perform basic user/group account management functions such that each group has their own administrator to manage common account issues and has full control on how their data/metadata needs to be annotated, shared and controlled in a consistent manner. This will also alleviate the burden for a system administrator without much domain knowledge about each DOC(group)’s business operations.

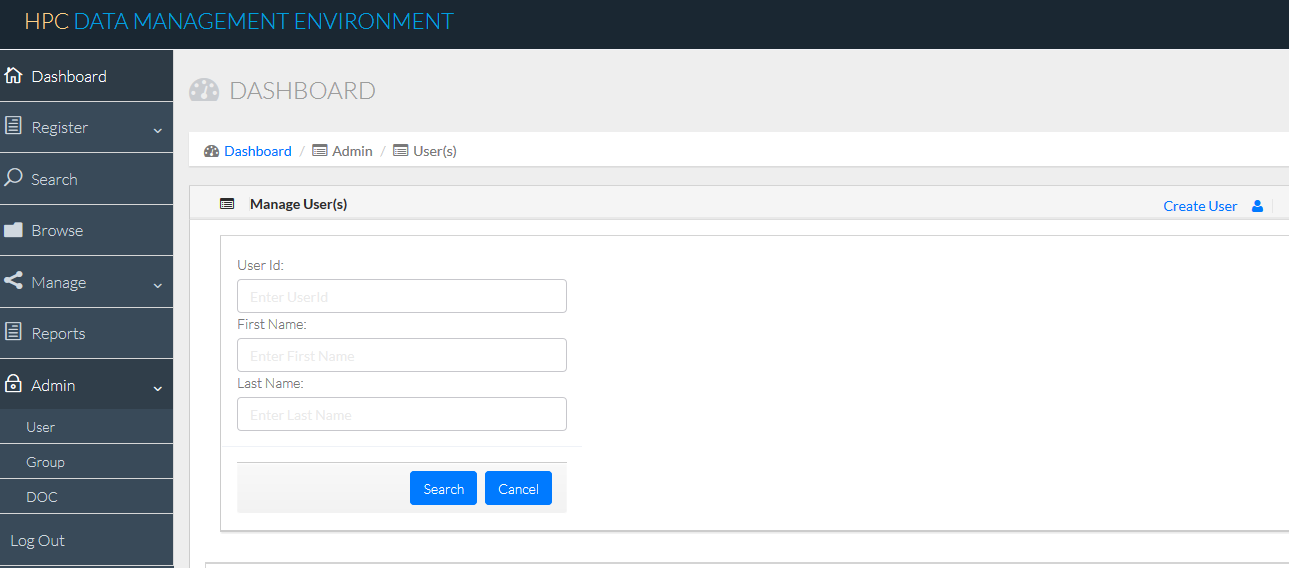


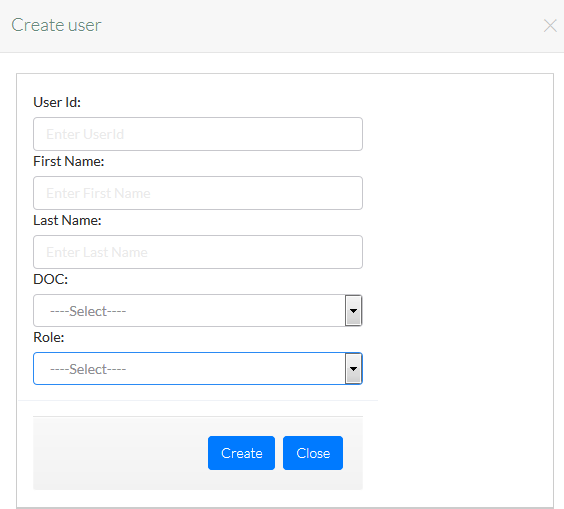
### User

A Group Administrator accesses the Admin User menu to add a new user for his/her group.

The Create user action can be triggered by clicking the hyperlink “Create User” on Admin Manage User(s) page. On the following user entry form, the administrator wants to ensure that:

1. NIH login user ID is entered for the User Id text entry field
2. The user’s actual first name is entered for First Name text entry
3. The user’s actual last name is entered for Last Name text entry
4. The actual DOC (group) value will be pre-populated on the DOC dropdown using the DOC the administrator is associated with
5. For the Role dropdown, either USER or “GROUP\_ADMINISTRATOR” may be selected or entered depending on the actual need





An administrator can search existing users using the Search button. All common search features/capabilities are supported.

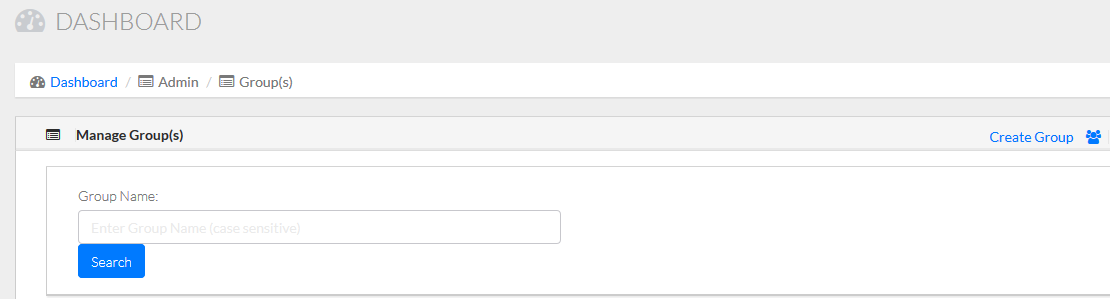
### Group

A Group Administrator accesses the Admin Group menu to add a new user group for his/her own DOC (Lab/Group/Operational Unit) business operational needs. The action of Create group

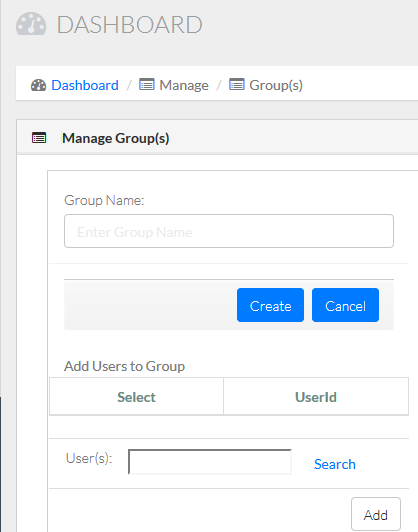
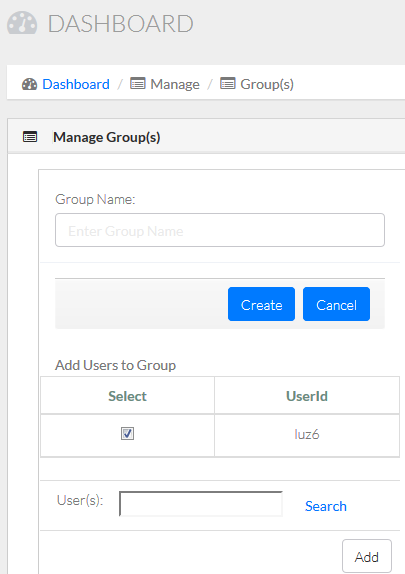
can be triggered by clicking the hyperlink “Create Group” on Admin Manage Group(s) page. On the following group entry form, the administrator wants to ensure that a meaningful user group name is determined and entered. Please note that the group name is not allowed to have empty space between words but may be concatenated with “\_” or “-“. A user group name of “Dummy test” with a pace in between “Dummy” and “test” will not be accepted; However, name entry of “Dummy\_test” will be accepted and transacted. Moreover, a user group name will be unique and will not be allowed to be edited once it is created. Therefore, the Update function will be solely for changing membership associated with the group.

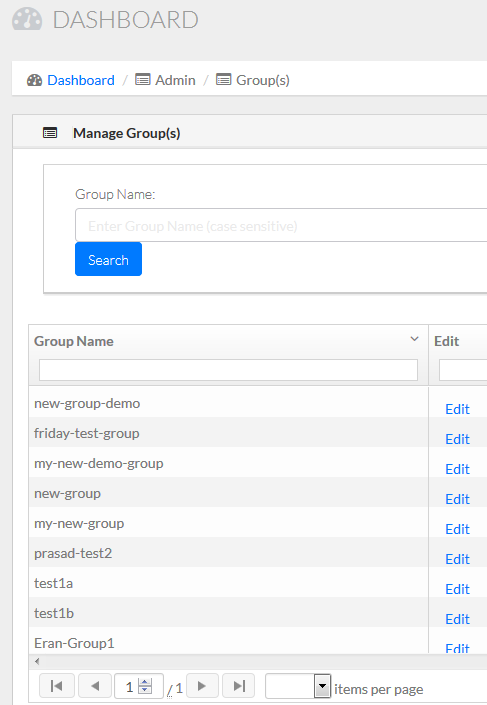
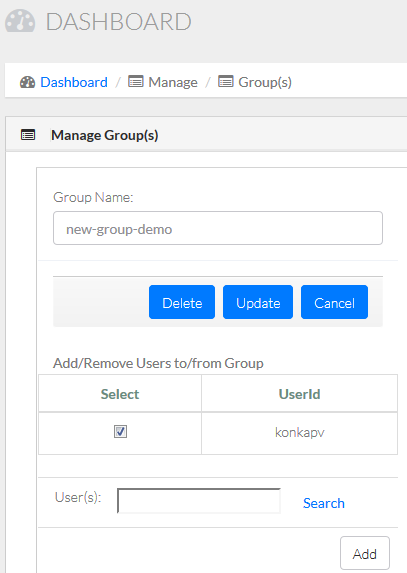
To maximize the benefits of adopting this feature, creation of a user group needs to be focused on the common authorizations desired to be in place on a group level. It will be advisable to create a user group, grant read access to this group, and then add individual users to the group if an administrator wants to share read access to some external staff/collaborators.

1. When creating a user group, an administrator may add the new group without associating a user member. However, s/he may add a user or users to the new group and create the group in one shot.
2. To add/remove a user or users to an existing user group, the administrator will search and locate the Group to edit first. When editing the Group selected, s/he will search and assign all interested users first to the grid list under “***Add/Remove Users to/from Group***”. Lastly, the Administrator will check/uncheck the checkbox adjacent to the user(s) to be associated/removed from the Group. A click on the Update button then will commit the transaction to update the group membership eventually.



Similarly, an administrator can search existing user groups using the Search button. All common search features/capabilities are supported.

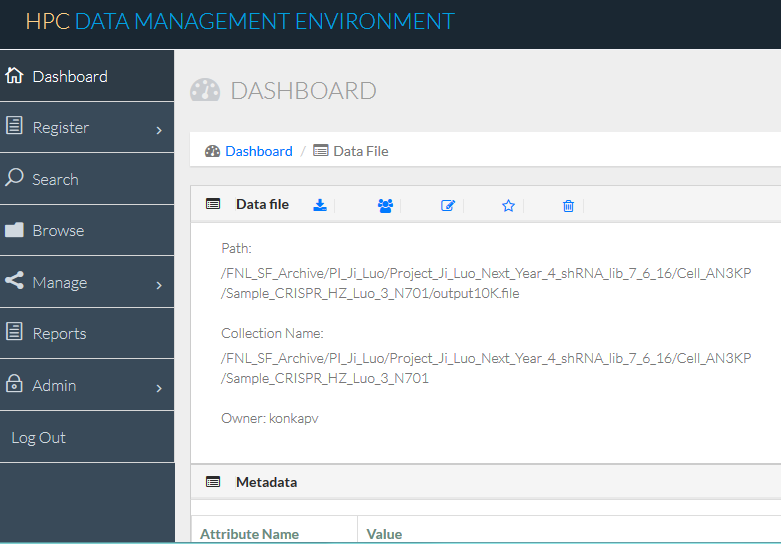
 

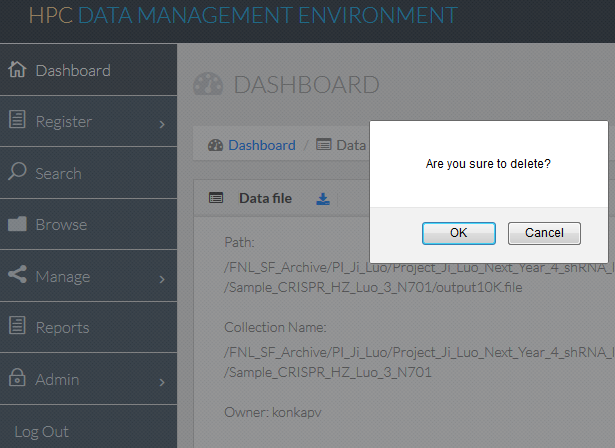
 

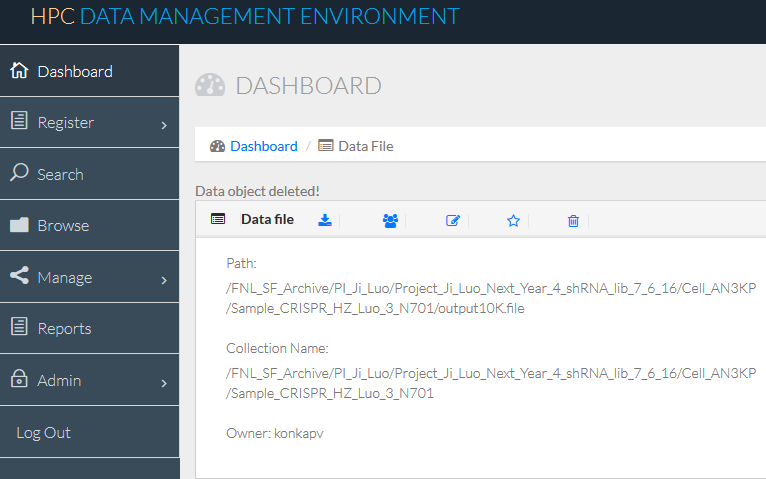
### Delete a Data Object and Associated Metadata

Sometimes though occurring not frequently, a user may end up uploading the wrong data files into the Archive. There would be a need to delete these data files and associated metadata while the integrity and such usage is exercised to the minimum extent. At this point, we will give such delete privilege to the Group Administrator role only. When a group admin selects this delete, the data object from Archive and all the associated metadata from iRODS backend database will be deleted physically. Therefore, a group administrator is advised to perform such delete operation only when s/he is absolutely certain that such data files are entered in by mistake and need to be deleted permanently.

The action of Create group can be triggered by clicking the most right “Delete” icon on the top of [Dashboard](https://fr-s-hpcdm-gp-d.ncifcrf.gov/dashboard)/Data File page for the selected data object to be deleted permanently. On the following confirmation dialog, the group administrator will select “OK” to proceed or “Cancel” to back out. After confirming the operation, the Web GUI performs the transaction via the HPC DME API and indicates that data object deleted with the collection (directory), file path, and owner presented on the screen.







# APPENDIX A – sample collection input

[https://github.com/CBIIT/HPC\_DME\_APIs/tree/master/utils/hpc-client/samples/Batch\_Upload\_Input\_File\_Collections.csv](https://ncisvn.nci.nih.gov/svn/HPC_Data_Management/branches/hpc-prototype-dev/src/hpc/hpc-irods-client/samples/Batch_Upload_Input_File_Collections.csv)

# APPENDIX B – sample data file input

[https://github.com/CBIIT/HPC\_DME\_APIs/tree/master/utils/hpc-client/samples/[Batch\_Upload\_Input\_File\_Object\_Cleversafe.csv](https://github.com/CBIIT/HPC_DME_APIs/blob/master/utils/hpc-client/samples/Batch_Upload_Input_File_Object_Cleversafe.csv)](https://ncisvn.nci.nih.gov/svn/HPC_Data_Management/branches/hpc-prototype-dev/src/hpc/hpc-irods-client/samples/Batch_Upload_Input_File_Object.csv)

[https://github.com/CBIIT/HPC\_DME\_APIs/tree/master/utils/hpc-client/samples/[Batch\_Upload\_Input\_File\_Object\_Globus.csv](https://github.com/CBIIT/HPC_DME_APIs/blob/master/utils/hpc-client/samples/Batch_Upload_Input_File_Object_Cleversafe.csv)](https://ncisvn.nci.nih.gov/svn/HPC_Data_Management/branches/hpc-prototype-dev/src/hpc/hpc-irods-client/samples/Batch_Upload_Input_File_Object.csv)

# Appendix c – sample permissions input

[https://github.com/CBIIT/HPC\_DME\_APIs/tree/master/utils/hpc-client/samples/Batch\_Permissions.csv](https://ncisvn.nci.nih.gov/svn/HPC_Data_Management/branches/hpc-prototype-dev/src/hpc/hpc-irods-client/samples/Batch_Permissions.csv)

# APPENDIX D – Executing HPC DME API with Soap UI

SoapUI is an open-source web service testing application for service-oriented architectures (SOA) and representational state transfers (REST). Its functionality covers web service inspection, invoking, development, simulation and mocking, functional testing, load and compliance testing.

### 13.1 Prerequisites

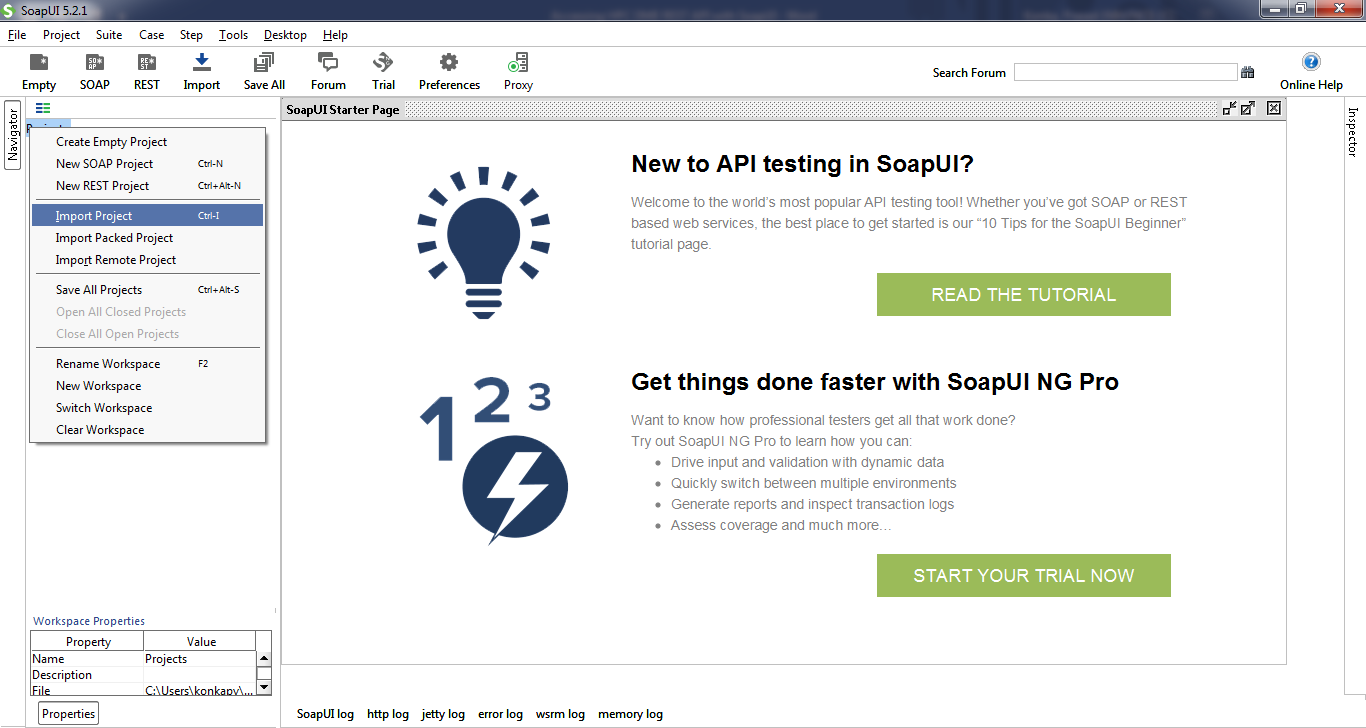
If you do not have SoapUI installed, please go to following link to download and install. <https://www.soapui.org/downloads/soapui.html>

Go to: <https://ncisvn.nci.nih.gov/svn/HPC_Data_Management/branches/hpc-prototype-dev/doc> and naviage to /training/ hpc\_dme\_soapui\_tutorial. You should see following files:

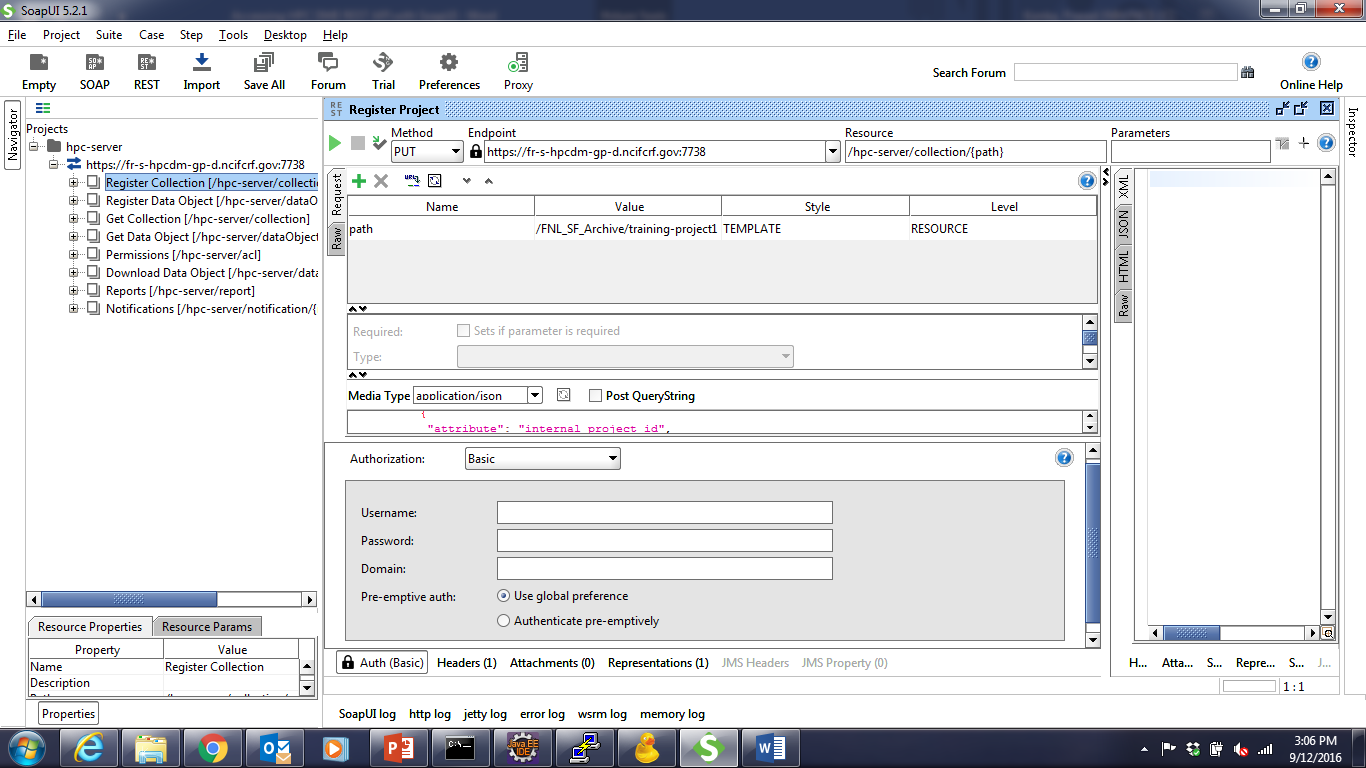
hpc-training-soapui-project.xml

* dataRegistration-async.json
* dataRegistration-sync.json
* SRR062635.filt.fastq

Once the SoapUI installation is complete, open SoapUI tool and select “Import Project” from File menu. Select “hpc-training-soapui-project.xml”.



Once the project is imported, you will see the project in the navigator. This project is created with resources to access HPC DME REST interfaces. The training project is prepopulated with multiple endpoint to access. Please select any endpoint that you have access to.

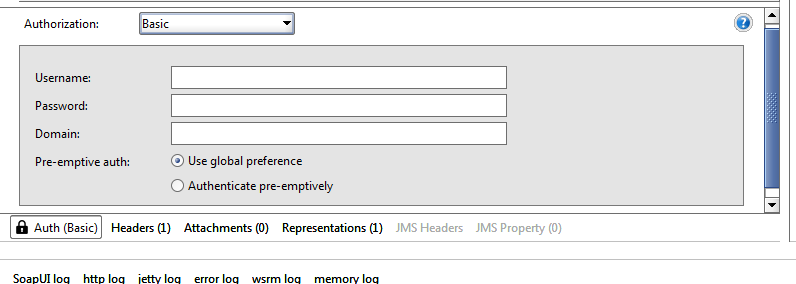


Additionally, valid NCI UserId and password are needed to authenticate with HPC DME API. Please contact HPC DME administrator (A system admin email address?) to get access on training application

### 13.2 Access HPC DME with Soap UI

To access HPC DME API, you will need to provide NCI UserId and Password through “Basic” authentication method. As shown the picture below, click on “Authentication and Security related settings” button. Select “Add New authorization” and select “Basic”. This step has to be done for every API request you make from SoapUI. Once you set it up for a request, you don’t have to recreate “Authentication and Security related settings” again.



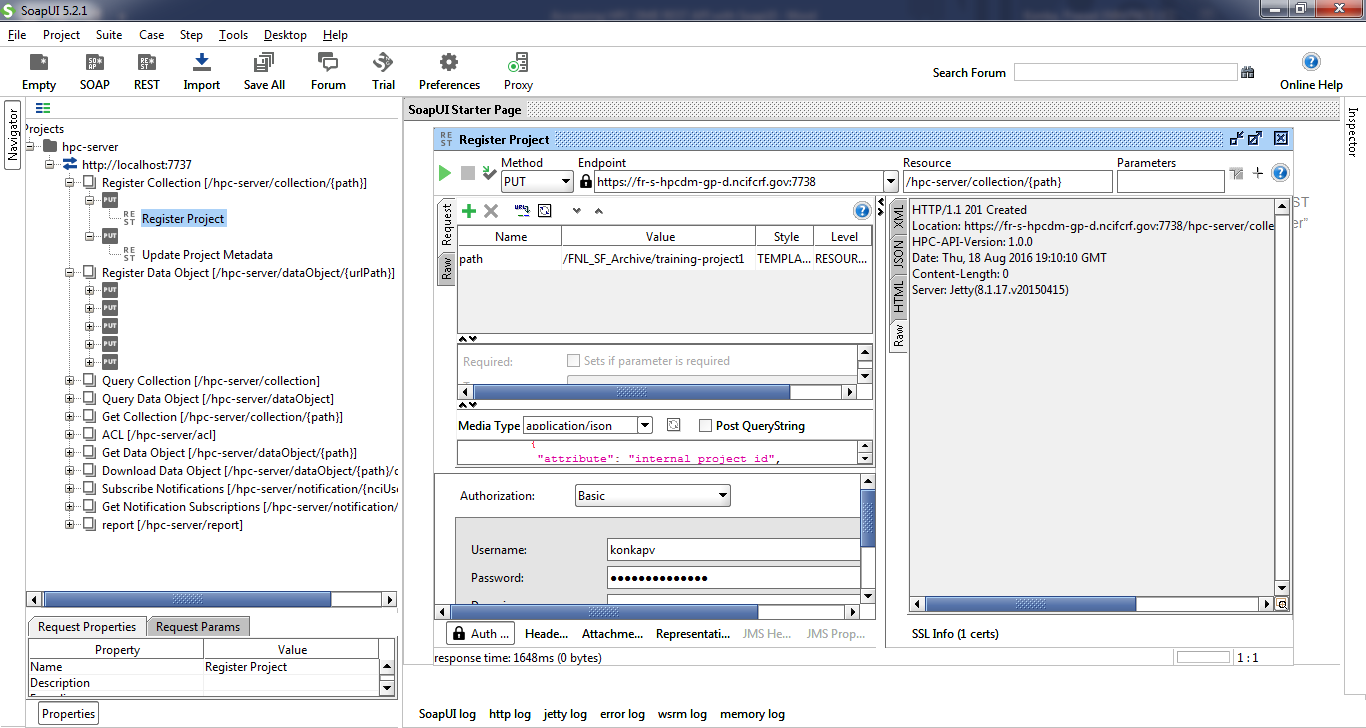


### 13.3 Use SOAP UI to perform common use scenarios

When registering collections and data files, one must keep in mind how the data and metadata are organized in their own division or subgroup and the order data and hierarchic are managed in the logic that needs to be enforced. For instance, if the structure of “Project/Dataset/Data file” needs to be implemented, then the parent collection “Project” must be created (registered) first along with the required metadata. The subcollection “Dataset” must be created next along with that level of metadata. Only then data files under the specific dataset and project can be registered in the correct sequence.

## Register a collection (Project or dataset)

“Project” is one of the collection types setup on the training endpoint. Expand “Register Collection” node, “PUT” and double click on “Register Project”. Make sure select training endpoint from the endpoint drop down. In the request section, edit “Value” column to enter desired project path. Path is a logical unique identifier to refer to a collection. For example, “/FNL\_SF\_Training/konkapv/Project1”. Click on triangle icon to submit request. You shall see the response from “Raw” tab.

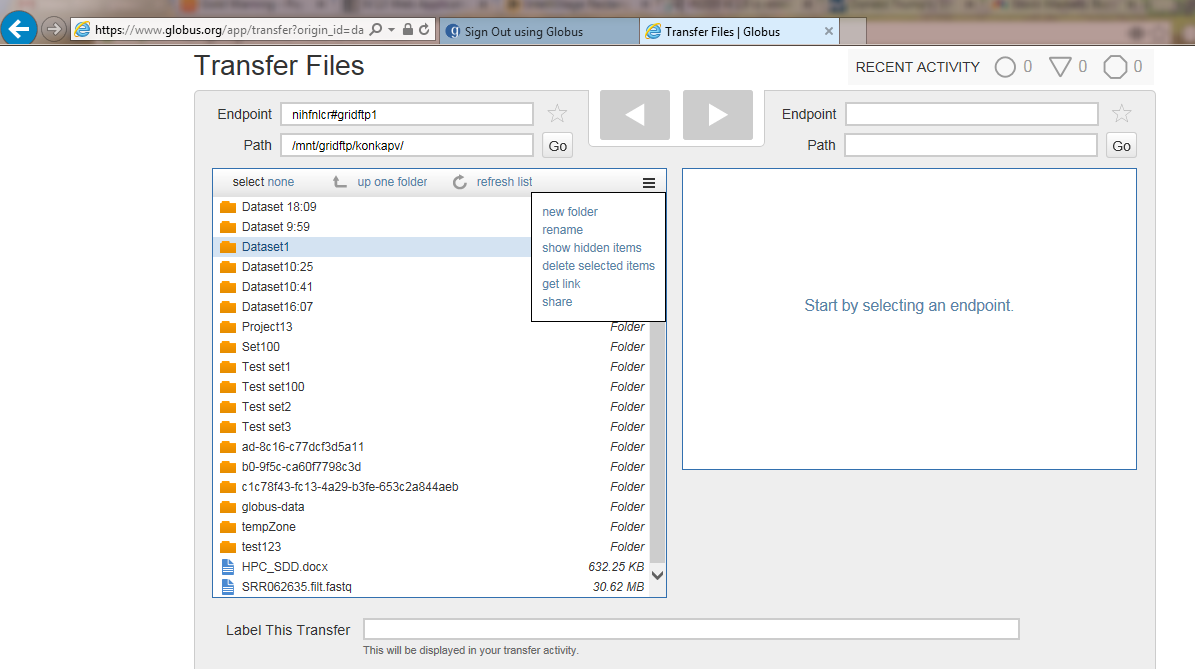


Dataset is the subcollection of project (a child of project) in our sample implementation. Double click on “Register Dataset”. Make sure to select training endpoint from the endpoint drop down. In the request section, edit “Value” column to enter desired dataset path. This path should be child of an existing project. For example, “/FNL\_SF\_Training/konkapv/Project1/Dataset1”. “Click on triangle icon to submit request. You shall see the response from “Raw” tab.

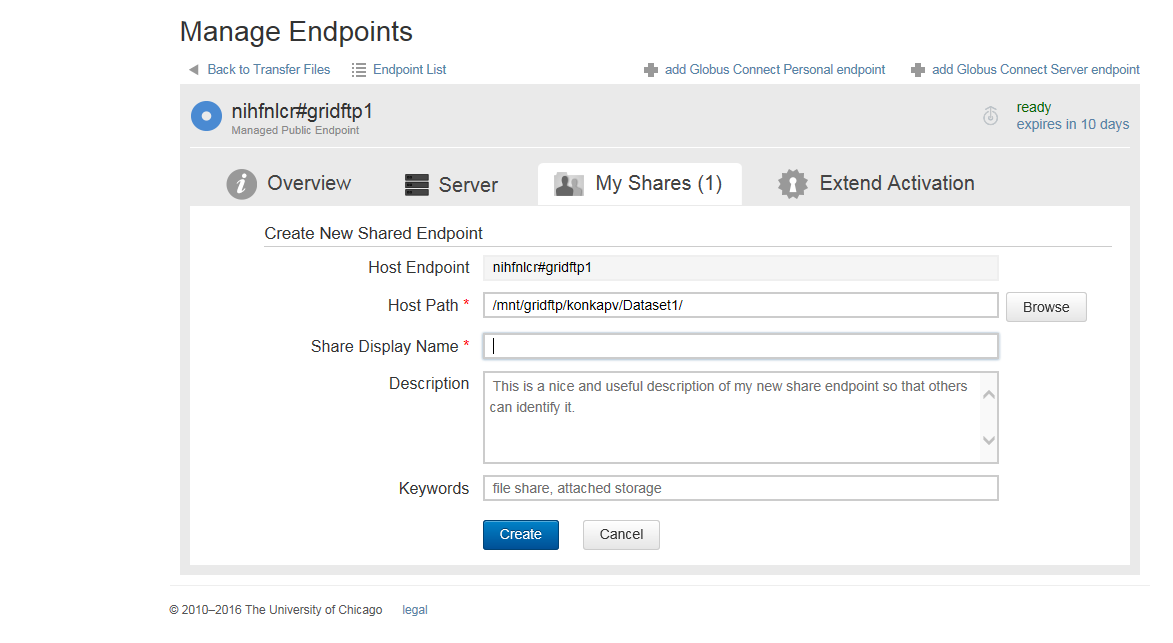


## Register a single data file/object into storage archive asynchronously

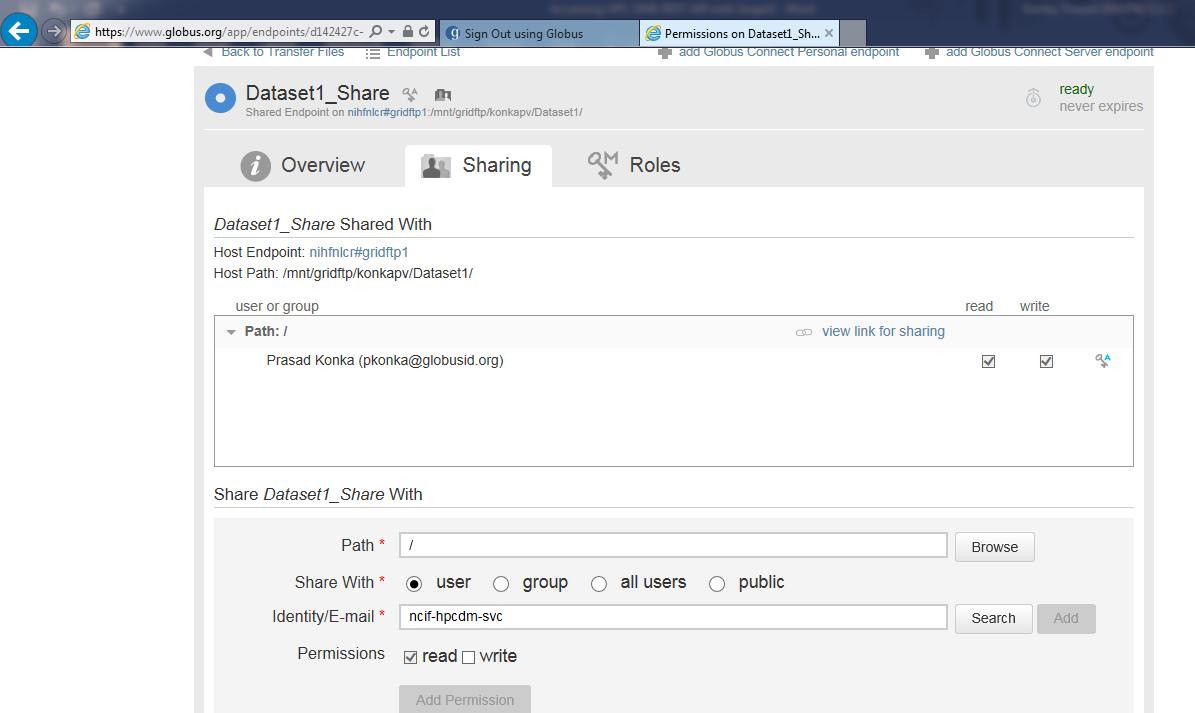
Prerequisite to register data object asynchronously is to have the data source location at a Globus endpoint with shared access to “ncif-hpcdm-svc” account. To create a shared access, login into [www.globus.org](http://www.globus.org) and select the endpoint where your data is located. Select the folder, and click on the icon on top right corner of the section. Click on “Share”.



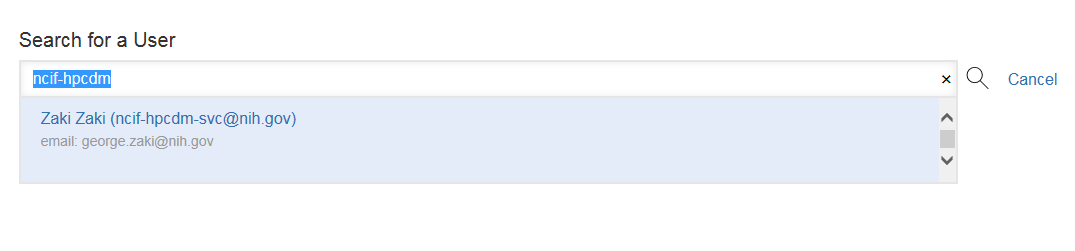
Enter “Share Display Name” and click on “Create” button.



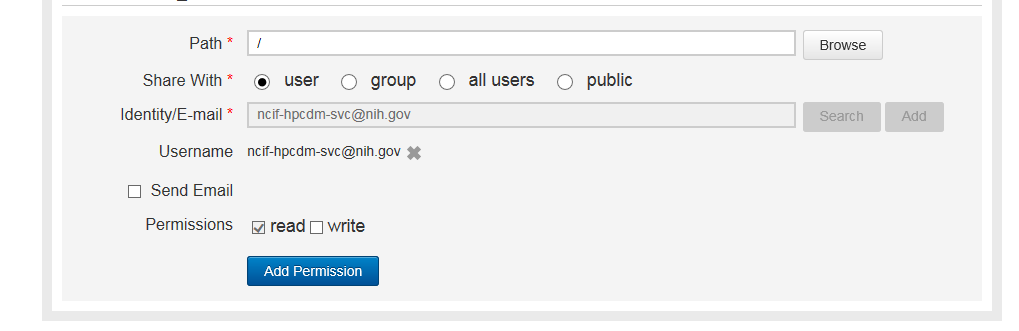
On data sharing page, select “User” radio button and click on “Search” button.



On “Search for a User” dialog, enter “ncif-hpcdm” and click on search icon. Select “Zaki Zaki” service user account.

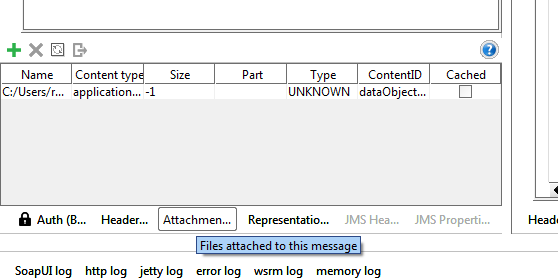


Uncheck “Send Email” and make sure “read” permission is selected. Click on “Add permission” button.



Data file registration is done via HTTP multipart request to HPC REST interface. Tutorial package has sample metadata file (dataRegistration-async.json) to register. Edit “dataRegistration-asyn.json” package to set source fileContainerId and source fileId. “fileContainerId” is the Globus UUID and “fileId” is the full path of the file exist on Globus.

Click on “attachments” tab and click on + sign shown below to add an attachment.



Select “dataRegistration-asyn.json” you edited. Once it is attached, double click on “Content type” and enter “application/json” as the value. Double click on “ContentId” and enter “dataObjectRegistration” as the value.

Double click request value to enter unique object path value. For example, “/FNL\_SF\_Training/konkapv/Project1/Dataset1/Object1”.

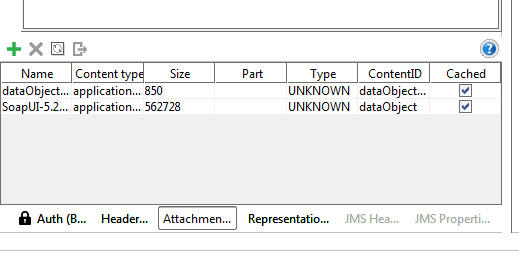
Click on triangle icon to submit request. You shall see the response from “Raw” tab.

## Register a single data file/object into storage archive synchronously

Data file registration is done synchronously via HTTP multipart request to HPC REST interface. Tutorial package has sample metadata file (dataRegistration-sync.json) to register. Edit “dataRegistration-syn.json” to update any metadata.

Attach metadata to the request (multipart): a) ContentType: application/json; b) ContentID: dataObjectRegistration

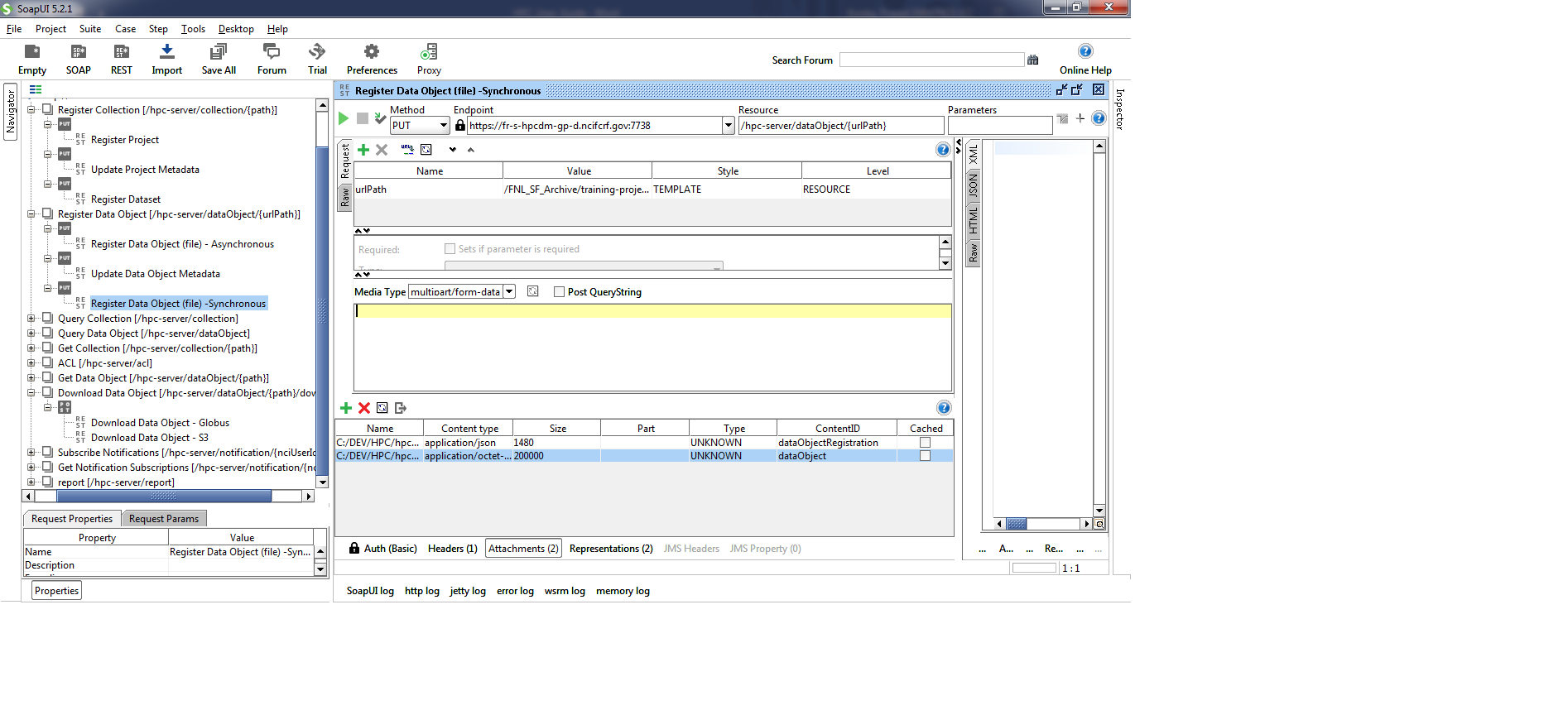
Click “Attachments” tab and click on the + sign shown below to add attachments mentioned above.

* + 1. 

Select “dataRegistration-sync.json” you edited. Double click on “Content type” and enter “application/json” as the value. Double click on “ContentId” and enter “dataObjectRegistration” as the value.

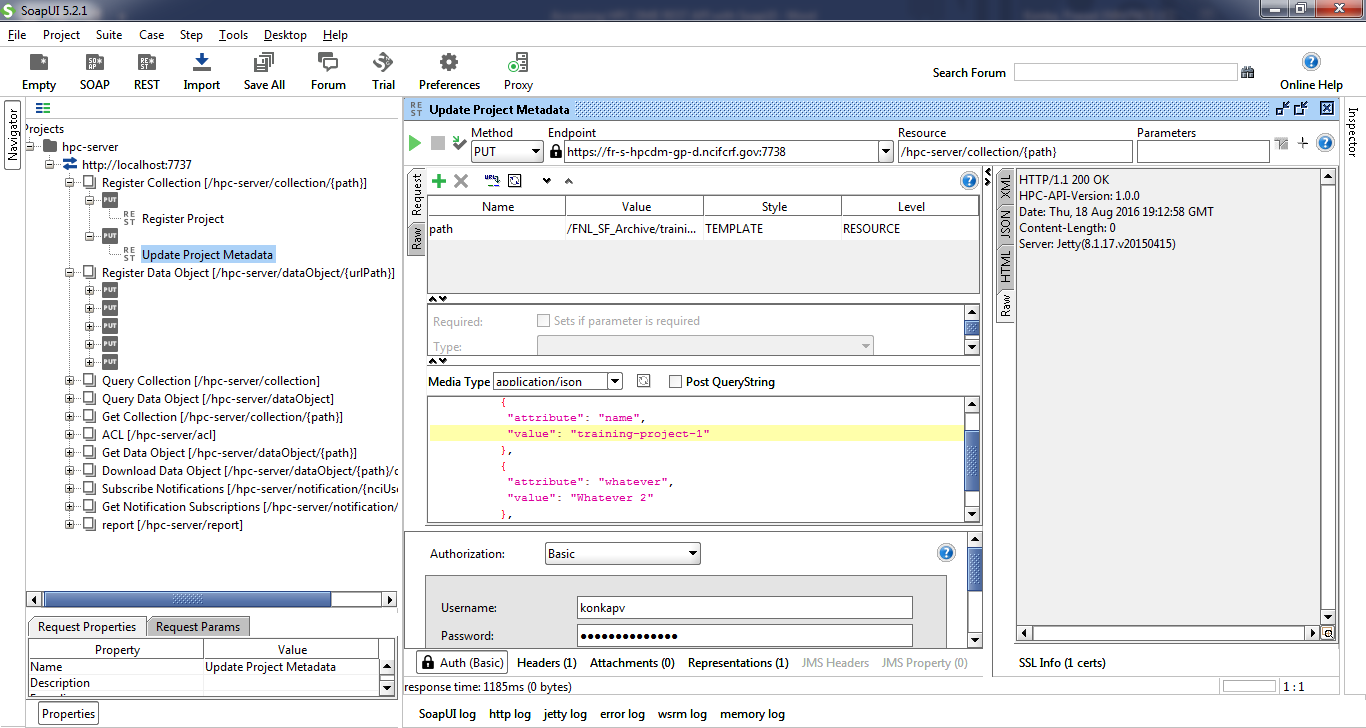
Click on the + sign shown below to add attachments mentioned above. Select data object file you want to register. Double click on “ContentId” and enter “dataObject” as the value. Double click request value to enter unique object path value.

Click on triangle icon to submit request. You shall see the response from “Raw” tab.



## Perform update on a metadata attribute

An existing Collection or data file metadata can be updated to add new metadata attributes or update existing metadata attribute value. We demonstrate how to perform this using the “Project” collection. Open “Update Project Metadata” resource and make sure request attribute value shows any existing project path. For example, “/FNL\_SF\_Training/konkapv/Project1”. You may edit metadata input information from media window. Click on triangle icon to submit request. You shall see the response from “Raw” tab.



## Subscribe to a known event

You may subscribe to get notification emails to the events generated during data upload and download requests. This subscription is to get notification on the requests you initiate. You can also unsubscribe from some of the notifications you are already subscribed to.

Following are the valid event notifications:

DATA\_TRANSFER\_UPLOAD\_IN\_TEMPORARY\_ARCHIVE:

DATA\_TRANSFER\_UPLOAD\_ARCHIVED:

DATA\_TRANSFER\_UPLOAD\_FAILED:

DATA\_TRANSFER\_DOWNLOAD\_COMPLETED:

DATA\_TRANSFER\_DOWNLOAD\_FAILED:

Edit request media content as needed. Click on triangle icon to submit request. You shall see the response from “RAW” tab.



## Generate a report

Authorized users can generate following summarized reports

Summary report till date

Summary report by date

Summary report by DOC

Summary report by DOC and data range

Summary report by User

Summary report by User and data range

Valid report types are:

USAGE\_SUMMARY

USAGE\_SUMMARY\_BY\_DATE\_RANGE

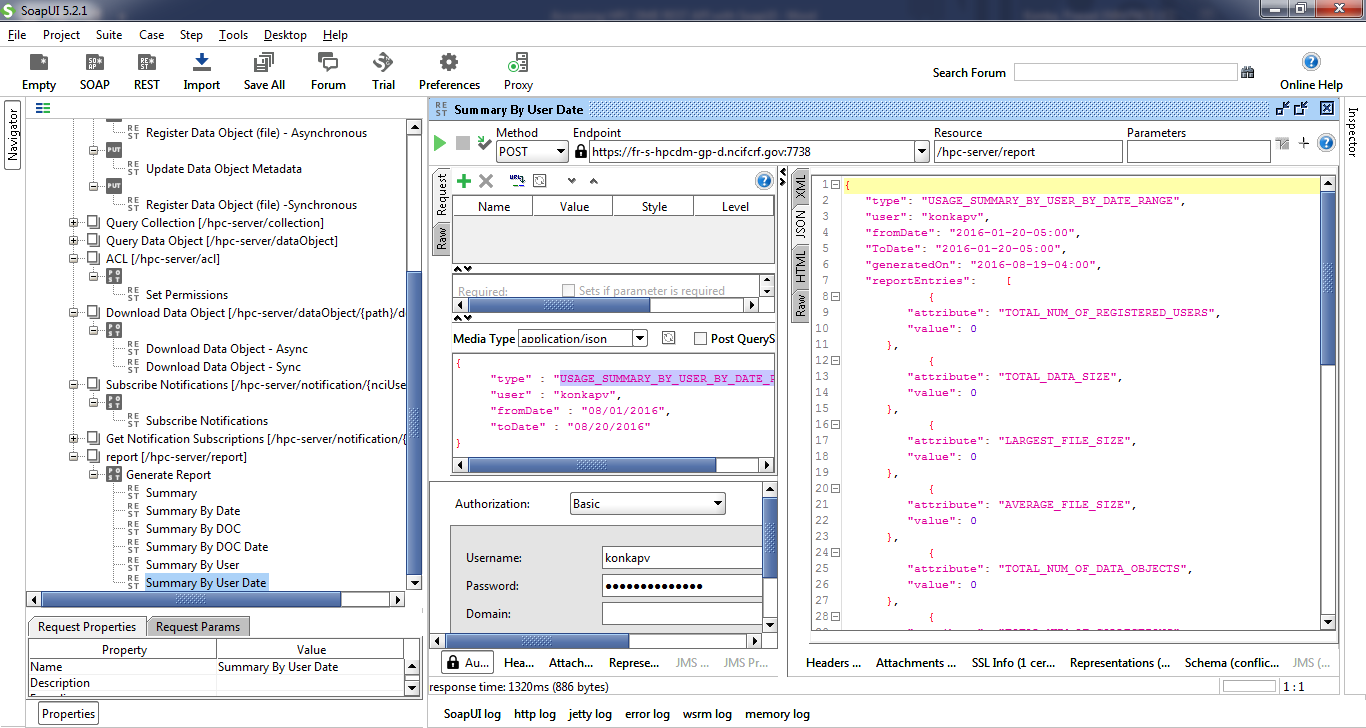
USAGE\_SUMMARY\_BY\_DOC

USAGE\_SUMMARY\_BY\_DOC\_BY\_DATE\_RANGE

USAGE\_SUMMARY\_BY\_USER

USAGE\_SUMMARY\_BY\_USER\_BY\_DATE\_RANGE

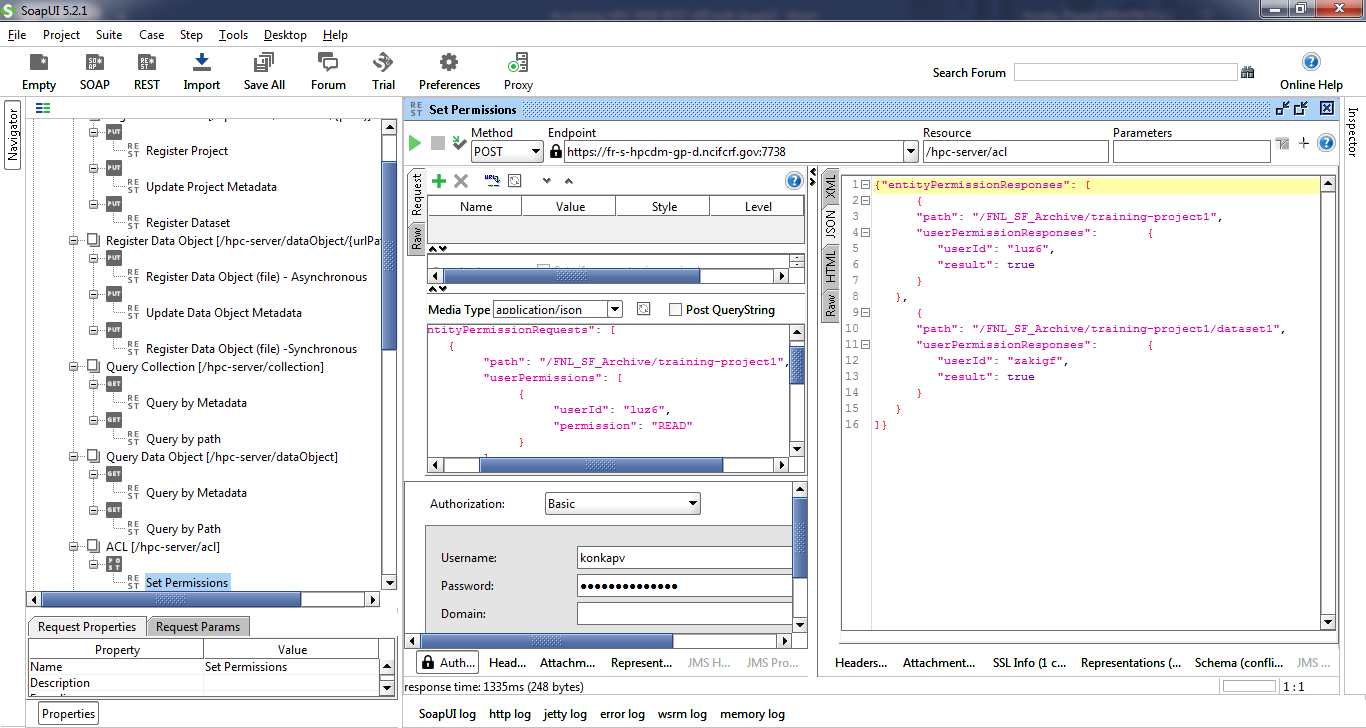
Edit request media content to enter valid request JSON request. Click on triangle icon to submit request. You shall see the response from “JSON” tab.



## Update/assign permission

You can set permissions on the collections or data objects so that any other authenticated user can access your data. Supported permissions are OWN, READ, WRITE and NONE.

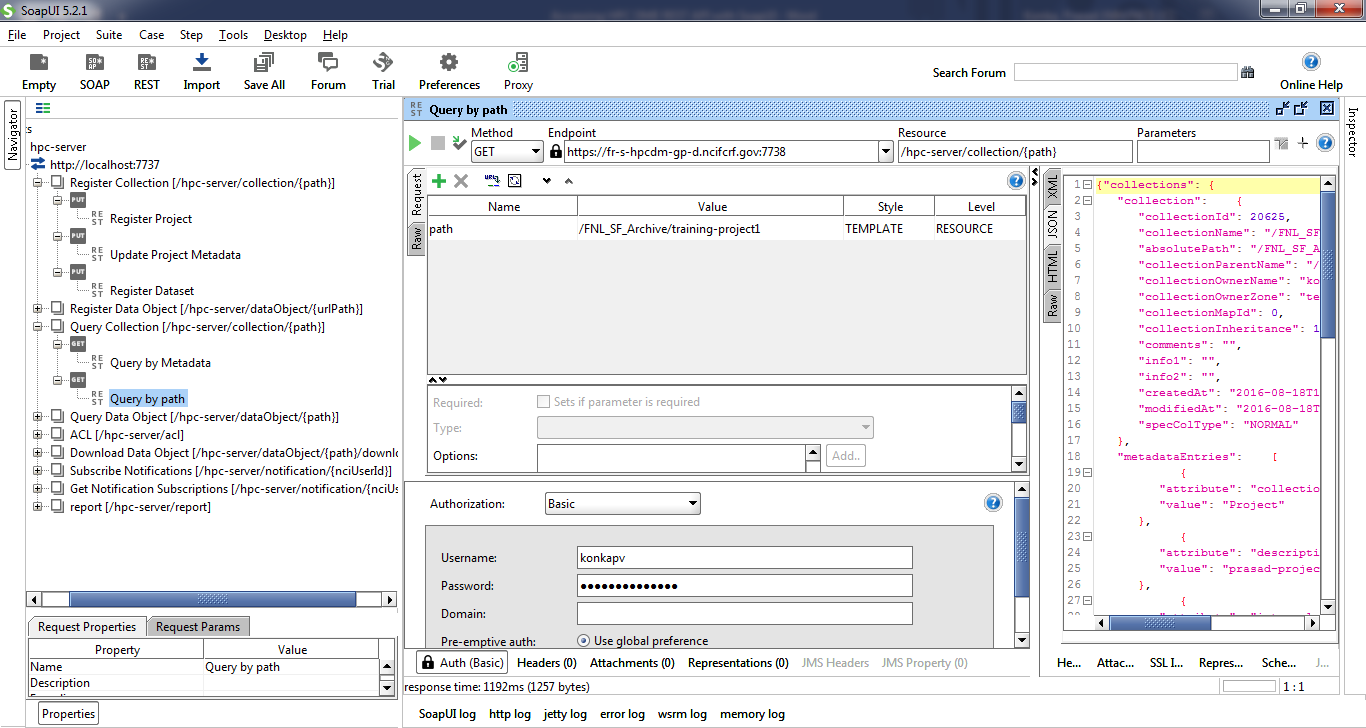
These permissions can be associated with a user or a group and are set by entity PATH. This entity can be a collection or a data file. Edit request media string to enter value entities and permissions. Click on triangle icon to submit request. You shall see the response from “JSON” tab.



## Perform simple search functions

##### 13.3.8.1 Query a collection by path

Collections can be queried by its unique logical path. Double click on request path attribute value and enter a valid collection path. For example, “/FNL\_SF\_Training/konkapv/Project1”. Click on triangle icon to submit request. You shall see the response from “JSON” tab.



##### 13.3.8.2 Query a collection by metadata

Collections can be queried by their metadata. Due to the limitation of adding multiple values to a parameter within SOAP UI, you could only search by one metadata attribute. Double click on metadataQuery request attribute value field and enter search criteria.

Format is

{"a":"<attribute name>","v":"<Value>","o":"<Operator>"}

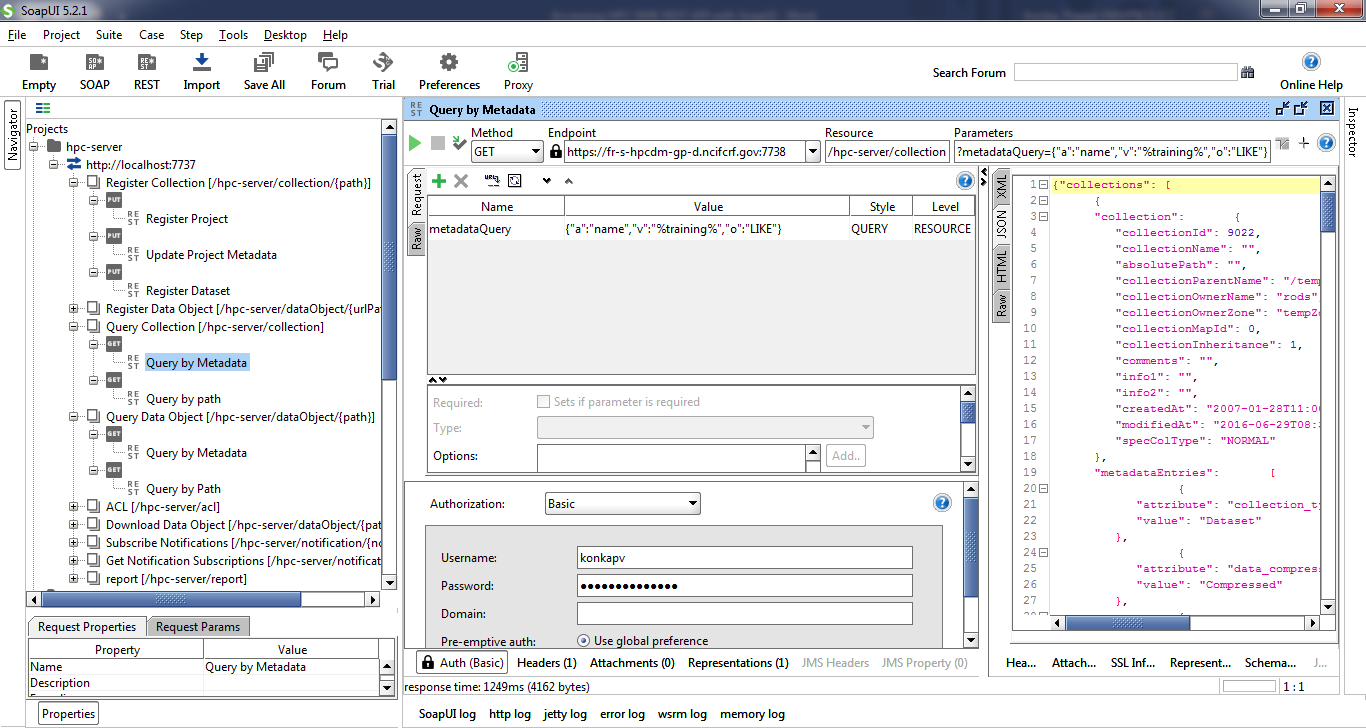
Replace <text> with the actual values. Valid values for <Operator> are

EQUAL, NOT\_EQUAL, LESS\_THAN, GREATER\_THAN, LIKE

Example:

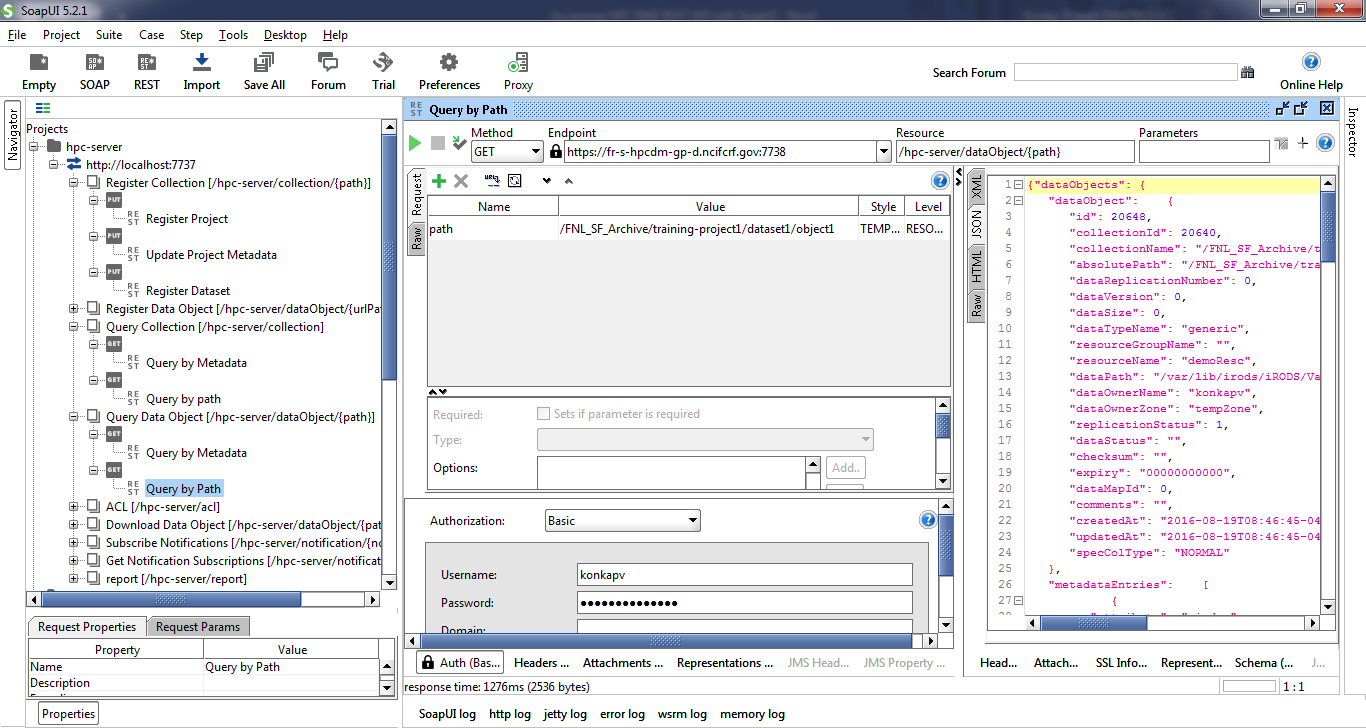
{"a":"name","v":"%training%","o":"LIKE"}

Click on triangle icon to submit request. You shall see the response from “JSON” tab.



##### 13.3.8.3 Get a data object by path

Data objects can be queried by its unique logical path. Double click on request path attribute value and enter a valid data object path. Click on triangle icon to submit request. You shall see the response from “JSON” tab.



##### 13.3.8.4 Get a data object by metadata

Data objects can be queried by their metadata. Due to the limitation of adding multiple values to a parameter, you could only search by one metadata attribute. Double click on metadataQuery request attribute value field and enter search criteria.

Format is

{"a":"<attribute name>","v":"<Value>","o":"<Operator>"}

Replace <text> with the actual values. Valid values for <Operator> are

EQUAL, NOT\_EQUAL, LESS\_THAN, GREATER\_THAN, LIKE

Example:

{"a":"name","v":"%object%","o":"LIKE"}

Click on triangle icon to submit request. You shall see the response from “JSON” tab.

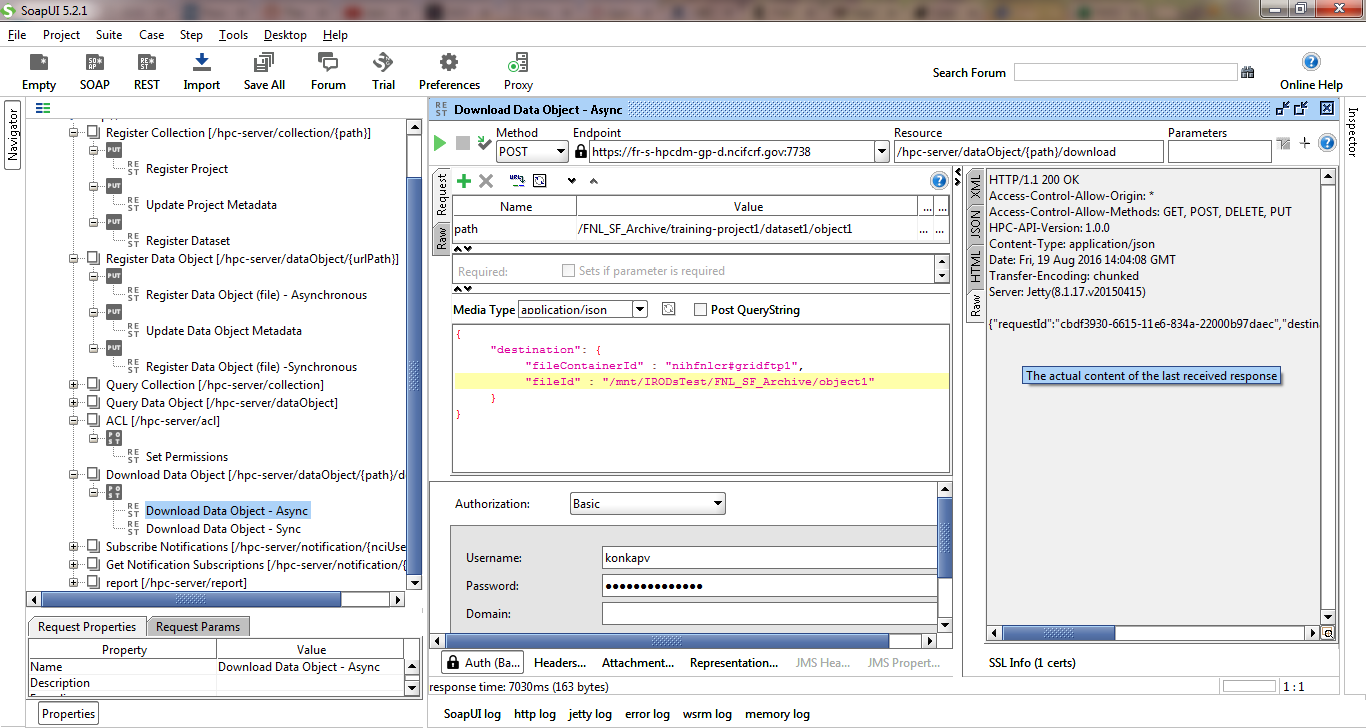


## Download a data file/object to Globus share

You can download a data object that you have access to into your Globus endpoint location asynchronously. The destination Globus endpoint needs be shared with “ncif-hpcdm-svc” account to write, i.e., you need to give “write” access on shared location to the Globus application account. Double click on request path attribute value and enter valid object path.

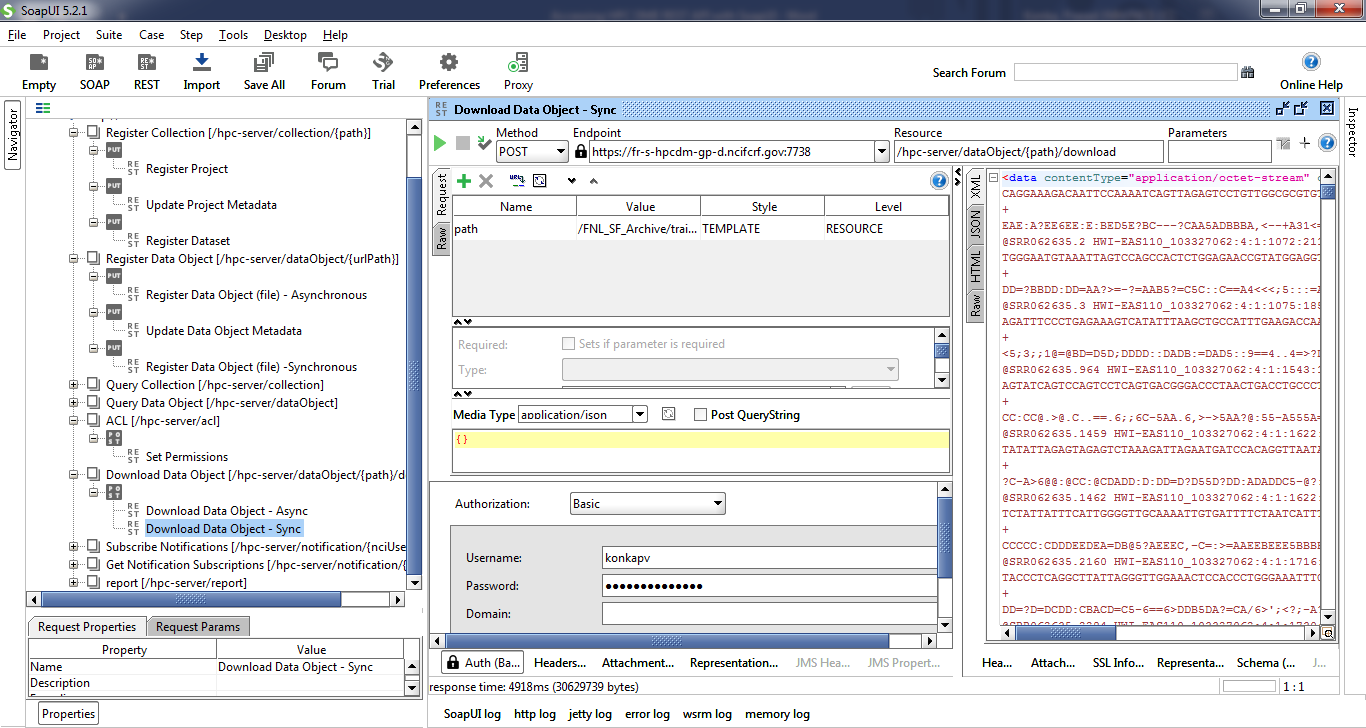
Edit destination values as needed.

Click on triangle icon to submit request. You shall see the response from “JSON” tab.



## Download a data file/object to a local directory

You can download from the archive to your local file system synchronously. Double click on request path attribute value and enter valid object path. Click on triangle icon to submit request. You shall see the response from “JSON” tab.



# APPENDIX E HPC DME CLI properties

|  |  |  |
| --- | --- | --- |
| Property name | Description | Default value |
| hpc.server.url | HPC Server API url. This is where REST interfaces are running on the server | <https://hpcdmeapi.nci.nih.gov> |
| hpc.default.globus.endpoint | Globus endpoint to transfer from |  |
| hpc.collection.service | Collection service REST resource name. You don’t need to change this value unless it is changed on the server. | collection |
| hpc.dataobject.service | Data object REST resource name. You don’t need to change this value unless it is changed on the server. | dataObject |
| hpc.error-log.dir | Path of the error log file to be written | . |
| hpc.ssl.keystore.path | Path of the keystore used for 2-way SSL connectivity with HPC Server. If the value is missing, one way SSL connection is used. | hpc-client/keystore/keystore-uat.jks |
| hpc.ssl.keystore.password | Password for the keystore. You don’t have to change this value. If the value is missing, one way SSL connection is used. | hpc-server-store-pwd |
| hpc.login.credentials | Location of the file with user credentials (user name and password). Credentials should be in format of <userId:password> |  |
| hpc.job.thread.count | Number of concurrent threads to process batch input file. Setting up large number of threads may not necessarily improve processing time. It depends on various factors like client machine processing speed, network bandwidth, server response. | 5 |
| globus.nexus.url | Globus Nexus URL. You don’t have to change this property value. | nexus.api.globusonline.org |
| globus.url | Globus URL for asynchronous data registration. You don’t have to change this property value. | [www.globusonline.org](http://www.globusonline.org) |
| hpc.login.token | HPC USER login token file location. You don’t have to change this property value. | tokens/hpcdme-auth.txt |
| hpc.globus.login.token | HPC USER globus login token file location. You don’t have to change this property value. | tokens/globus-auth.txt |