HPC Data MANAGEMENT Environment

General Training

---------------------------------------------------------------------------------------

Version Number: 1.1

Version Date: 11/2/2016

VERSION HISTORY

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version Number** | **Implemented**  **By** | **Revision**  **Date** | **Approved**  **By** | **Approval**  **Date** | **Description of Change** |
| 1.0 | Zhengwu Lu | 8/18/2016 |  |  | Initial Draft |
| 1.1 | Prasad Konka | 11/2/2016 |  |  | Curl examples |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

TABLE OF CONTENTS

[1 Introduction 4](#_Toc465860963)

[1.1 Purpose of the Training material 4](#_Toc465860964)

[1.2 users & assumptions 4](#_Toc465860965)

[1.3 Estimated Storage Need & Resource allocation 4](#_Toc465860966)

[2 Security 5](#_Toc465860967)

[2.1 Authentication 5](#_Toc465860968)

[2.2 Authorization 5](#_Toc465860969)

[2.3 Assumptions / Constraints 6](#_Toc465860970)

[3 POLICies 6](#_Toc465860971)

[3.1 Metadata definition and logical entities of hirarchies 6](#_Toc465860972)

[3.2 policy file definition 6](#_Toc465860973)

[3.3 Assumptions / Constraints 7](#_Toc465860974)

[4 Access HPC DME 7](#_Toc465860975)

[4.1 HPC Client utility 7](#_Toc465860976)

[4.2 Soap UI 8](#_Toc465860977)

[4.3 CURL 23](#_Toc465860978)

[Appendix A: Approval 29](#_Toc465860979)

[APPENDIX B: REFERENCES 30](#_Toc465860980)

# Introduction

## Purpose of the Training material

This general training reference material presents users the functional capabilities of the HPC data management environment (DME) APIs and show cases common use scenarios how a user may interact and use these APIs or command utility tool to accomplish their goals in dealing with scientific data management functions or daily tasks, such as performing a batch upload, metadata attribute update, or search function based on known metadata.

## users & assumptions

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.

Note: HPC DME supports only NCI account holders at this point of time.

For general training purpose in getting familiarized with the HPC DME APIs, we focus on using the batch utility tool to perform bulk upload functions, the SOAP UI tool to perform a single record type update/creation or simple search with some primitive user interface, the CURL open source command line tool and library for calling/executing DME APIs with URL syntax. The emphasis for this training material is on how to use the tools to perform common use functions via calling the HPC DME APIs.

HPC DME is accessible through its Service API or through its client interface such as Java, Perl, Ruby, Curl, JavaScript languages programmatically or through UI clients like SOAP UI, REST Client. Please refer to DME User Guide document for DME Service API link and verify the pre-requisites.

## Estimated Storage Need & Resource allocation

The HPC data management environment provides a number of application programming interfaces (APIs) to support common scientific data/metadata management functions.

The current DME APIs are deployed at UAT tier for 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. This will be helpful and used to guide the subsequent tier provisioning (Training, Production) to maintain system scalability.

# Security

## Authentication

HPC DME users are configured to get authenticated with NCI Active Directory credentials.

## Authorization

HPC DME supports the following roles:

1. USER – HPC DME user role with access to the following functions:
   * Update self-profile
   * Register a data object with metadata
   * Update metadata on data object
   * Download data object
   * Share data object
   * Search collections and data objects
   * Generate reports
2. GROUP\_ADMIN – HPC DME group admin role with access to following functions in addition to all USER role allowed functions:
   * List group info
   * Create a group
   * Create user
   * Add user to a group
   * Remove user from a group
   * Register a collection with metadata
   * Update collection metadata
   * Generate reports

c) SYSTEM\_ADMIN – HPC DME Super User role to manage users, roles, permissions, DOC (Division/Office/Center) definitions.

## Assumptions / Constraints

### Each user is registered under a unique DOC. A user will not belong to 2 or more DOCs

### A user in DOC A will not be able to view or update data attributes, that belong to DOC B

### Each DOC will have its own group administrator (s)

# POLICies

## Metadata definition and logical entities of hirarchies

### HPC DME supports metadata registration to connect unstructured data files or objects. Metadata may be attached to files, collections (Equivalent of sub-directories).

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.

Though majority of biomedical research facilities will adopt or customize their organization of data or metadata using the MIAMI notation, HPC DME does enable flexibility for each DOC to have their own way of organizing/structuring how their data/metadata is presented or preserved. To illustrate, we utilize the very simple hierarchical structure “Project/Dataset/Data file” to showcase how each level of the collections and the bottom level data files will be registered.

## policy file definition

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.

## Assumptions / Constraints

Policy files will be defined through each DOC or group with their individual policies considered and agreed. Policy files will be deployed to HPC DME server by DME system administrator

# Access HPC DME

## HPC Client utility

HPC DME Client Utility provides easy to access command line interface to interact with HPC DME APIs.

### Batch File Registration

### HPC DME User Guide has a designated section in detailing how to perform a batch file execution.

### A batch run will invoke APIs in the backend to register all orchestrated DOC organized data/metadata, data objects and specific collections and subcollections in metadata catalog and designated storage archive systematically. Should there be any errors occurring during in processing the batch operation, the commanding console will be shown: How many rows of data were written and how many were skipped; and an error log will be created to indicate the specifics for all skipped rows. Upon successful batch job runs, a user may use specific reports or the search APIs to verify or confirm if the data files interested have been archived. The commanding console will also indicate similar features

### Integrated Search

HPC DME User Guide section “HPC DME Batch Client” describes in detail how various search functions may be performed on either collections or data files (objects) level with the option of having the interim search results written into a csv/json/txt file

### The integrated search results with filter criteria applied may contain user indicated metadata and pointers to the associated data files in the archive. This capability will allow users to dive further into the data files and associated metadata if s/he is interested, hence facilitating user workflow in managing pipeline.

## Soap UI

### SOAP UI Overview

### 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.

### 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

### 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.





### 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

##### 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.



##### Query a collection by metadata

Collections 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":"%training%","o":"LIKE"}

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



##### 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.



##### 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 service 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.



## CURL

### CURL command overview

Curl is a command-line tool for transferring data using various protocols. It can be used to interact with the HPC DME REST API.

### Prerequisites

Make sure that 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 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.

Navigate to: [https://ncisvn.nci.nih.gov/svn/HPC\_Data\_Management/branches/hpc-prototype-dev/doc/training/curl\_training](curl_training) to download sample input files.

### Use CURL 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)

Please edit create\_project\_input.json file to have valid metadata attributes for your DOC.

|  |
| --- |
| Command:  Curl -k -H “Content-Type: application/json” -d @create\_project\_input.json -X PUT <**server:port**>/hpc-server/collection/<**collection-path**> -H "Accept: application/json" -u <**userId**>  Example:  curl -k -H "Content-Type: application/json" -d @create\_project\_input.json -X PUT https://fr-s-hpcdm-gp-d.ncifcrf.gov:7738/hpc-server/collection/FNL\_SF\_Archive/curl\_training/project\_1 -H "Accept: application/json" -u test\_user |

Please edit create\_dataset\_input.json file to have valid metadata attributes for your DOC.

|  |
| --- |
| Command:  Curl -k -H “Content-Type: application/json” -d @create\_dataset\_input.json -X PUT <**server:port**>/hpc-server/collection/<**dataset-path**> -H "Accept: application/json" -u <**userId**>  Example:  curl -k -H "Content-Type: application/json" -d @create\_dataset\_input.json -X PUT https://fr-s-hpcdm-gp-d.ncifcrf.gov:7738/hpc-server/collection/FNL\_SF\_Archive/curl\_training/project\_1/dataset\_1 -H "Accept: application/json" -u test\_user |

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

To register data file asynchronously through Globus, please refer to HPC User Guide – Section 4.4 (Data Transfer) for the pre-requisite.

Please edit create\_datafile\_async\_input.json file to have valid metadata attributes for your DOC.

|  |
| --- |
| Command:  Curl -k -H “Content-Type: application/json” -d @create\_datafile\_async\_input.json -X PUT <**server:port**>/hpc-server/dataObject/<**datafile-path**> -H "Accept: application/json" -u <**userId**>  Example:  curl -k -H "Content-Type: application/json" -d @create\_datafile\_async\_input.json -X PUT https://fr-s-hpcdm-gp-d.ncifcrf.gov:7738/hpc-server/dataObject/FNL\_SF\_Archive/curl\_training/project\_1/dataset\_1/file\_1 -H "Accept: application/json" -u test\_user |

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

Please edit create\_datafile\_sync\_input.json file to have valid metadata attributes for your DOC.

|  |
| --- |
| Command:  Curl -k -H “Content-Type: application/json” -d @create\_datafile\_sync\_input.json -X PUT <**server:port**>/hpc-server/dataObject/<**datafile-path**> -H "Accept: application/json" -u <**userId**>  Example:  curl -k -H "Content-Type: application/json" -d @create\_datafile\_sync\_input.json -X PUT https://fr-s-hpcdm-gp-d.ncifcrf.gov:7738/hpc-server/dataObject/FNL\_SF\_Archive/curl\_training/project\_1/dataset\_1/file\_2 -H "Accept: application/json" -u test\_user |

#### Perform update on a metadata attribute

Please edit update\_project\_input.json file to have valid metadata attributes for your DOC.

|  |
| --- |
| Command:  Curl -k -H “Content-Type: application/json” -d @update\_project\_input.json -X PUT <**server:port**>/hpc-server/collection/<**project-path**> -H "Accept: application/json" -u <**userId**>  Example:  curl -k -H "Content-Type: application/json" -d @create\_dataset\_input.json -X PUT https://fr-s-hpcdm-gp-d.ncifcrf.gov:7738/hpc-server/collection/FNL\_SF\_Archive/curl\_training/project\_1 -H "Accept: application/json" -u test\_user |

#### Subscribe to a known notification

|  |
| --- |
| Command:  Curl -k -H “Content-Type: application/json” -d @subscribe\_notifications\_input.json -X PUT <**server:port**>/hpc-server/ notification/<**userId**> -H "Accept: application/json" -u <**userId**>  Example:  curl -k -H "Content-Type: application/json" -d @subscribe\_notifications\_input.json -X PUT https://fr-s-hpcdm-gp-d.ncifcrf.gov:7738/hpc-server/ notification/test\_user -H "Accept: application/json" -u test\_user |

#### Generate a report

Please edit summary\_report\_user\_input.json file to enter your UserId.

|  |
| --- |
| Command:  Curl -k -H “Content-Type: application/json” -d @summary\_report\_user\_input.json -X PUT <**server:port**>/hpc-server/report -H "Accept: application/json" -u <**userId**>  Example:  curl -k -H "Content-Type: application/json" -d @summary\_report\_user\_input.json -X PUT https://fr-s-hpcdm-gp-d.ncifcrf.gov:7738/hpc-server/report -H "Accept: application/json" -u test\_user |

#### Update/assign permission

Please edit assign\_permission\_input.json file to enter UserId and desired permission. Valid permissions are “READ”, “WRITE”, “OWN”, “NONE”. Please refer to HPC API Specification for more details on security.

|  |
| --- |
| Command:  Curl -k -H “Content-Type: application/json” -d @ assign\_permission\_input.json -X PUT <**server:port**>/hpc-server/acl -H "Accept: application/json" -u <**userId**>  Example:  curl -k -H "Content-Type: application/json" -d @ assign\_permission\_input.json -X PUT https://fr-s-hpcdm-gp-d.ncifcrf.gov:7738/hpc-server/acl -H "Accept: application/json" -u test\_user |

#### Perform simple search functions

##### Query a collection by path

|  |
| --- |
| Command:  Curl -k -H “Content-Type: application/json” -X GET <**server:port**>/hpc-server/collection/<**PATH**> -H "Accept: application/json" -u <**userId**>  Example:  curl -k -H "Content-Type: application/json" -X PUT https://fr-s-hpcdm-gp-d.ncifcrf.gov:7738/hpc-server/collection/FNL\_SF\_Archive/curl\_training/project\_1 -H "Accept: application/json" -u test\_user |

##### Query a collection by metadata

Please edit criteria input files to suit your search criteria. You may limit your search criteria to one or more. All these criteria clauses are executed with AND operator.

|  |
| --- |
| Command:  Curl -k -H “Content-Type: application/json” -X GET <**server:port**>/hpc-server/collection --data-urlencode metadataQuery@<**criteria1**> --data-urlencode metadataQuery@<**criteria2**> --data-urlencode metadataQuery@<**criteria3**> -H "Accept: application/json" -u <**userId**>  Example:  curl -k -H "Content-Type: application/json" -X PUT <https://fr-s-hpcdm-gp-d.ncifcrf.gov:7738/hpc-server/collection> --data-urlencode metadataQuery@search\_project\_input1.json --data-urlencode metadataQuery@search\_project\_input2.json --data-urlencode metadataQuery@search\_project\_input3.json -H "Accept: application/json" -u test\_user |

##### Get a data object by path

|  |
| --- |
| Command:  Curl -k -H “Content-Type: application/json” -X GET <**server:port**>/hpc-server/dataObject/<**PATH**> -H "Accept: application/json" -u <**userId**>  Example:  curl -k -H "Content-Type: application/json" -X PUT https://fr-s-hpcdm-gp-d.ncifcrf.gov:7738/hpc-server/dataObject/FNL\_SF\_Archive/curl\_training/project\_1/dataset\_1/file\_1 -H "Accept: application/json" -u test\_user |

##### Get a data object by metadata

Please edit criteria input files to suit your search criteria. You may limit your search criteria to one or more. All these criteria clauses are executed with AND operator.

|  |
| --- |
| Command:  Curl -k -H “Content-Type: application/json” -X GET <**server:port**>/hpc-server/dataObject --data-urlencode metadataQuery@<**criteria1**> --data-urlencode metadataQuery@<**criteria2**> --data-urlencode metadataQuery@<**criteria3**> -H "Accept: application/json" -u <**userId**>  Example:  curl -k -H "Content-Type: application/json" -X PUT https://fr-s-hpcdm-gp-d.ncifcrf.gov:7738/hpc-server/dataObject --data-urlencode metadataQuery@search\_datafile\_input1.json --data-urlencode metadataQuery@search\_datafile2\_input2.json --data-urlencode metadataQuery@search\_datafile3\_input3.json -H "Accept: application/json" -u test\_user |

#### Download a data file/object to Globus share

Please edit download\_async\_input.json to enter your destination globus endpoint shared with “ncif-hpcdm-svc” user.

|  |
| --- |
| Command:  Curl -k -H “Content-Type: application/json” -d @download\_async\_input.json -X GET <**server:port**>/hpc-server/dataObject/<**PATH**>/download -H "Accept: application/json" -u <**userId**>  Example:  curl -k -H "Content-Type: application/json" -d @download\_async\_input.json -X PUT https://fr-s-hpcdm-gp-d.ncifcrf.gov:7738/hpc-server/dataObject/FNL\_SF\_Archive/curl\_training/project\_1/dataset\_1/file\_2/download -H "Accept: application/json" -u test\_user |

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

Please do not edit download\_sync\_input.json. This file has an empty json input.

|  |
| --- |
| Command:  Curl -k -H “Content-Type: application/json” -d @download\_sync\_input.json -X GET <**server:port**>/hpc-server/dataObject/<**PATH**>/download -H "Accept: application/json" -u <**userId**>  Example:  curl -k -H "Content-Type: application/json" -d @download\_sync\_input.json -X PUT https://fr-s-hpcdm-gp-d.ncifcrf.gov:7738/hpc-server/dataObject/FNL\_SF\_Archive/curl\_training/project\_1/dataset\_1/file\_1/download -H "Accept: application/json" -u test\_user |

Appendix A: Approval

The undersigned acknowledge that they have reviewed the ***NCI HPC Data Management Environment General Training*** and agree with the information presented within this document. Changes to this **Requirements Definition** will be coordinated with, and approved by, the undersigned, or their designated representatives.

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: |  | Date: |  |
| Print Name: |  |  |  |
| Title: |  |  |  |
| Role: |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: |  | Date: |  |
| Print Name: |  |  |  |
| Title: |  |  |  |
| Role: |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: |  | Date: |  |
| Print Name: |  |  |  |
| Title: |  |  |  |
| Role: |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: |  | Date: |  |
| Print Name: |  |  |  |
| Title: |  |  |  |
| Role: |  |  |  |

APPENDIX B: REFERENCES

The following table summarizes the documents referenced in this document.

|  |  |  |
| --- | --- | --- |
| **Document Name** | **Description** | **Location** |
| *HPC DME User Guide* | *User Guide* | *https://ncisvn.nci.nih.gov/svn/HPC\_Data\_Management/branches/hpc-prototype-dev/doc/guides/HPC\_User\_Guide.pdf* |
| *HPC DME API Specification* | *API Specification* | *https://ncisvn.nci.nih.gov/svn/HPC\_Data\_Management/branches/hpc-prototype-dev/doc/guides/HPC\_Server\_API.pdf* |