**High Performance Computing (HPC) Data Management Environment (DME) Getting Started**

**Purpose**

This document is intended to serve as a guide for getting started on using the High Performance Computing (HPC) Data Management Environment (DME) for the National Cancer Institute (NCI). Specifically, this guide is intended to help the reader perform registrations of data files into the DME using Bash shell utilities or Curl from Command line. In this context, bulk means multiple files per registration transaction.

**Assumptions**

Reader is comfortable: working with Bash shell command-line environment (or Cygwin Bash emulator on Windows), making minor edits to properties files, and decompressing archive files (using any tool of reader’s choice).

Based on user DOC policies set by DOC group administrator, reader is expected to be knowledgeable about the structure of the data and its metadata policies. If there are any metadata policies, Reader is comfortable generating metadata in json or csv format. If a reader intends to use Curl, their DOC has a structured data hierarchy and well-defined metadata policies and s/he is comfortable working with file scripting utilities to write, read and convert csv data into json format. (Optional) If reader intends to use Globus with DME, it is assumed they are familiar with Globus or can figure out how to perform various tasks with Globus without guidance from this document. For background information about Globus and how it can be used with DME, please read the DME [User Guide](https://github.com/CBIIT/HPC_DME_APIs/blob/master/doc/guides/HPC_User_Guide.docx) at GitHub..

**Prerequisites**

1. Java runtime environment (JRE)

To test if you have JRE installed and ready to use, in command window, type “java -version”. If it is installed, command returns version information about JRE.

1. Curl 7.5.4 or above utility

To test if you have Curl instance and ready to use, in the command window, type “curl --version”. If it is installed, command returns version information about Curl.

**Instructions on Setting Up HPC DME Client Utilities**

1. From GitHub, checkout the release branch (releases/<version number>) of the HPC\_DME\_APIs repository at <https://github.com/CBIIT/HPC_DME_APIs>. For e.g., if the release number is 1.6.0, then run the command

git clone -b releases/1.6.0  https://github.com/CBIIT/HPC\_DME\_APIs.git

1. Cd to ./utils directory, and rename or copy hpcdme.properties-sample to hpcdme.properties.
2. Edit *hpcdme.properties* file to make configuration changes. Sub-items a-d below offer elaboration about this. It is recommended to make backup copy of this file before editing.
   1. Uncomment the UAT Server settings or the Production Server settings depending on which environment you want to connect to. By default, the UAT server settings are enabled.
   2. Set the *hpc.ssl.keystore.password* property to correct value. Request this password from appropriate HPC DME technical point of contact.
   3. Change *hpc.user* property to your username (a.k.a. your NIH username).
   4. Optionally for Globus integration, uncomment and change *hpc.default.globus.endpoint* property to UUID of Globus endpoint to utilize.
   5. For other properties not mentioned above, only change settings if you are confident you know what you are doing. Otherwise, leave them as-is.
3. For Cygwin users only, run following command to alter *functions* file to use Unix standard end of line delimiters. $ sed -I "s/\r$//" functions
4. Edit your *~/.bashrc* or *~/.profile* file to append following lines in Courier New font but replacing <CLIENT\_UTLS\_HOME> with absolute path to that directory.

# export environment variable HPC\_DM\_UTILS pointing to directory where

# HPC DME client utilities are, then source functions script in there

export HPC\_DM\_UTILS=<CLIENT\_UTILS\_HOME>

source $HPC\_DM\_UTILS/functions

1. Source whichever file you modified in Step 5 to make the changes take effect in current Bash session. $ source ~/.bashrc Or $ source ~/.profile

**How to Register Data Files with HPC DME Client Utilities**

Preliminary steps – Do these prior to running utilities that perform DME transactions

1. Open or switch to a command window.
2. Run DME client utility *dm\_generate\_token* to generate DME API authentication token. With a token, you avoid repeatedly typing your password for every command requiring authentication to DME API server. $ dm\_generate\_token

The subsequent table describes DME client utilities you can use to register/put multiple files in DME using one transaction; in other words, the utilities support bulk registration. Please refer to the [User Guide](https://github.com/CBIIT/HPC_DME_APIs/raw/master/doc/guides/HPC_User_Guide.docx) at GitHub, In particular, the section "Executing HPC DME API with Command Line Utilities".

|  |  |
| --- | --- |
| **Summary: DME Client Utilities for Bulk Registration** | |
| **Utility** | **Description & Usage Example** |
| dm\_register\_directory | Register a Collection to DME using local directory.  Usage example:  $ dm\_register\_directory /home/joeuser/projX \  $ /someDOC/joeuser/projX  Notes:   1. First argument is path to local directory. 2. Second argument is destination path in DME. 3. Command as shown above selects specified directory and all its contents. 4. There are additional options not shown in above usage example which support finer-grained control. Refer to User Guide mentioned above this table for details. 5. To register metadata for the resulting Collection, there must be a JSON metadata file that is located in the same parent directory as the local directory. That file must have name that is the name of the local directory appended with file extension of ".metadata.json". 6. To register metadata for any resulting Sub Collection or Data Object in the resulting Collection, there must be a JSON metadata file that is in the same parent directory as the originating sub directory or file in the local file system. The metadata file must have name that is the corresponding item's name including file extension, if applicable, appended with the file extension of ".metadata.json". |
| dm\_register\_globus\_directory | Register a Collection to DME using a Globus directory.  Usage example:  $ dm\_register\_globus\_directory \  $ /myWork/studyXyz/ \  $ /theDOC/studies/studyXyz  Notes:   1. First argument is path to Globus directory at default Globus endpoint.    1. Default Globus endpoint is set in the properties file as described in the User Guide section “**Instructions on Setting Up HPC DME Client Utilities”** 2. Second argument is destination path in DME. 3. If there is no default Globus endpoint configured or you wish to use a Globus endpoint different from the default one, you may specify a third argument to indicate which Globus endpoint. This optional third argument must be UUID of intended Globus endpoint. |

**How to Register Data Files with Curl**

A DOC or lab may have a LIMS system to manage/track data files metadata information. However, registering such data files in bulk may be scripted even without an existing LIMS system at all.

**Following are the steps to write a script that registers collections:**

1. Extract collections metadata information into a CSV file. Apart from collection metadata attributes, each collection record in the CSV file must have “collection\_type” and “collection\_path” attributes. These attributes are defined and created to capture required and optional data elements/properties per each DOC domain specific and data sharing needs on the organizing structure (folder/subfolder) level. Please [click here](https://github.com/CBIIT/HPC_DME_APIs/blob/master/doc/training/samples/Batch_Register_Collections_Input_File.csv) for sample CSV file.
2. Read the CSV file and iterate through each row. Convert the CSV row into a JSON file. Please [click here](https://github.com/CBIIT/HPC_DME_APIs/blob/master/doc/training/samples/collection_metadata_input.json) for sample JSON file.
3. Get authentication token using the following curl command and write the token value into “auth\_token.cfg” file. Please [click here](https://github.com/CBIIT/HPC_DME_APIs/blob/master/doc/training/samples/auth-token.cfg) for sample authentication config file.

curl -k -G -X GET https://<hpc dme api server>/authenticate --user <UserId> -H "Accept: application/json" -o token.txt -D response.txt

*Example:* curl -k -G -X GET https://fr-s-hpcdm-gp-d.ncifcrf.gov:7738/hpc-server/authenticate --user jdoe -H "Accept: application/json" -o token.txt -D response.txt

1. For each collection row, use the following curl command to register the collection

curl -k -H "Content-Type: application/json" -d @<metadata>.json -X PUT https://<hpc dme api server>/collection/<Collection path> --config <Auth token config> -H "Accept: application/json"

Ex: curl -k -H "Content-Type: application/json" -d @collection\_metadata\_input.json -X PUT https://fr-s-hpcdm-gp-d.ncifcrf.gov:7738/hpc-server/collection/FNL\_SF\_Archive/PI\_Prasad --config auth-token.cfg -H "Accept: application/json"

1. (Optional) After successful return, delete JSON file from the step 2 above
2. (Optional) If there is an error response, write CSV record into a CSV file to try it again later.

**Following are the steps to write a script that registers data files:**

1. Extract data files metadata information into a CSV file. Apart from data file metadata attributes, each data file record in the CSV file must have “object\_path” and “file\_path” attributes. “file\_path” represents physical location of the file and “object\_path” is the logical path. Again, these attributes are defined and created to capture required and optional data elements/properties per each DOC domain specific and data sharing needs on each data file level.

Please [click here](https://github.com/CBIIT/HPC_DME_APIs/blob/master/doc/training/samples/Batch_Register_Sync_Dataobjects_Input_File.csv) for a sample CSV file representing synchronous registration. To register data files from Globus, CSV file should have “fileContainerId” and “fileId”. “fileContainerId” represents Globus endpoint UUID and “fileId” represents file location on Globus endpoint. Globus endpoint must be shared with HPC DME Application account with READ permission. Please check the User Guide for details. Please [click here](https://github.com/CBIIT/HPC_DME_APIs/blob/master/doc/training/samples/Batch_Register_Async_Dataobjects_Input_File.csv) for a sample CSV file representing asynchronous registration.

1. Read the CSV file and iterate through each row. Convert the CSV row excluding “file\_path” value into a JSON file. Please [click here](https://github.com/CBIIT/HPC_DME_APIs/blob/master/doc/training/samples/datafile_metadata_sync_input.json) for sample JSON file.
2. Get authentication token using the following curl command and write the token value into “auth\_token.cfg” file. Please [click here](https://github.com/CBIIT/HPC_DME_APIs/blob/master/doc/training/samples/auth-token.cfg) for sample authentication config file.

curl -k -G -X GET https://<hpc dme api server>/authenticate --user <UserId> -H "Accept: application/json" -o token.txt -D response.txt

*Example:* curl -k -G -X GET https://fr-s-hpcdm-gp-d.ncifcrf.gov:7738/hpc-server/authenticate --user jdoe -H "Accept: application/json" -o token.txt -D response.txt

1. For each data file row, use the following curl command to register the data file

curl -k -F "dataObjectRegistration=@<metadata>.json;type=application/json" -F "dataObject=@<data file>;type=application/octet-stream" -X PUT https://<hpc dme api server>/dataObject/<data file logical path> --config auth-token.cfg -H "Accept: application/json"

*Ex:* curl -k -F "dataObjectRegistration=@create\_dataobject\_input.json;type=application/json" -F "dataObject=@ERR0000861filt.fastq.gz;type=application/octet-stream" -X PUT https://fr-s-hpcdm-gp-d.ncifcrf.gov:7738/hpc-server/dataObject/FNL\_SF\_Archive/PI\_DrRied/ProjectX1/RunC8RCAACXX/SampleA1/ERR0000861filt --config auth-token.cfg -H "Accept: application/json"

1. (Optional) After successful return, delete JSON file from the step 2 above
2. (Optional) If there is an error response, write CSV record into a CSV file to try it again later.

[Click here](https://github.com/CBIIT/HPC_DME_APIs/blob/master/doc/training/samples/ArchiveProject_Prod_v1.3.pl) for a sample Perl script that registers collections and data files.

**Verifying Data Files registration**

You can use HPC DME web GUI client to examine Collections (directories/folders) and Data Objects (files) that are in HPC DME storage. For production web client, go to <https://hpcdmeweb.nci.nih.gov>. For UAT web client, go to <https://fr-s-hpcdm-uat-p.ncifcrf.gov/>. Information on how to use HPC DME web client is available in the [User Guide](https://github.com/CBIIT/HPC_DME_APIs/raw/master/doc/guides/HPC_User_Guide.docx) at GitHub, specifically the section "Executing HPC DME API with Web Client".