*<High Performance Computing Prototype>*

Requirements Definition Document

Version Number: 1.1

Version Date: 06/09/2017

***Notes to the Author***

[This document is a template of a Requirements Definition document for a project. The template includes instructions to the author, boilerplate text, and fields that should be replaced with the values specific to the project.

* Blue italicized text enclosed in square brackets ([text]) provides instructions to the document author, or describes the intent, assumptions and context for content included in this document.
* Blue italicized text enclosed in angle brackets (<text>) indicates a field that should be replaced with information specific to a particular project.
* Text and tables in black are provided as boilerplate examples of wording and formats that may be used or modified as appropriate to a specific project. These are offered only as suggestions to assist in developing project documents; they are not mandatory formats.

**When using this template, the following steps are recommended:**

1. Replace all text enclosed in angle brackets (e.g., <Project Name>) with the correct field document values. These angle brackets appear in both the body of the document and in headers and footers. To customize fields in Microsoft Word (which display a gray background when selected) select File->Properties->Summary and fill in the appropriate fields within the Summary and Custom tabs.

After clicking OK to close the dialog box, update all fields throughout the document selecting Edit>Select All (or Ctrl-A) and pressing F9. Or you can update each field individually by clicking on it and pressing F9.

These actions must be done separately for any fields contained with the document’s Header and Footer.

1. Modify boilerplate text as appropriate for the specific project.
2. To add any new sections to the document, ensure that the appropriate header and body text styles are maintained. Styles used for the Section Headings are Heading 1, Heading 2 and Heading 3. Style used for boilerplate text is Body Text.
3. To update the Table of Contents, right-click on it and select “Update field” and choose the option - “Update entire table”.
4. Before submission of the first draft of this document, delete this instruction section “Notes to the Author” and all instructions to the author throughout the entire document.

VERSION HISTORY

This requirements definition document will be updated continuously during the HPC prototyping effort following the iterative agile approach. The table below will be used to track the update and approval history of this document, person responsible and date of the task performed.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version Number** | **Implemented**  **By** | **Revision**  **Date** | **Approved**  **By** | **Approval**  **Date** | **Description of Change** |
| 0.1 | *KanakaDurga Addepalli* | 04/14/2015 |  |  | 1. Changed Table of Contents 2. Added information to Section 1, 2, 3 |
| 0.2 | *KanakaDurga Addepalli* | 05/05/2015 |  |  | Added Requirements |
| 0.3 | *KanakaDurga Addepalli* | 05/07/2015 |  |  | Added Requirements |
| 0.4 | *KanakaDurga Addepalli* | 05/12/2015 |  |  | Added and Updated Requirements |
| 0.5 | *KanakaDurga Addepalli* | 05/07/2015 |  |  | Added and Updated Requirements |
| 0.6 | *KanakaDurga Addepalli* | 05/018/2015 |  |  | Added and Updated Requirements |
| 0.7 | *KanakaDurga Addepalli* | 05/26/2015 |  |  | Added and Updated Requirements |
| 0.8 | *KanakaDurga Addepalli* | 05/28/2015 |  |  | Updated Requirements |
| 0.9 | *KanakaDurga Addepalli* | 06/01/2015 |  |  | Updated Requirements |
| 1.0 | *Zhengwu Lu* | 06/21/2016 |  |  | Revise and add additional requirements for supporting hierarchical searches and metadata attribute level authorizations |
| 1.1 | *Zhengwu Lu* | 06/09/2017 |  |  | Revise and enhance for GitHub migration |

TABLE OF CONTENTS

[Introduction 5](#_Toc484760965)

[1.1 Purpose of the Requirements Definition Document 5](#_Toc484760966)

[Business Requirements Overview 5](#_Toc484760967)

[1.2 Assumptions / Constraints 5](#_Toc484760968)

[1.3 System Scope 5](#_Toc484760969)

[Functional Requirements 7](#_Toc484760970)

[1.4 User Login and Logout 7](#_Toc484760971)

[1.5 User Home page/Dashboard 7](#_Toc484760972)

[1.6 Collection Registration 7](#_Toc484760973)

[1.7 File Registration 8](#_Toc484760974)

[1.8 Data Transfer 9](#_Toc484760975)

[1.9 Bulk Transfer of datasets/ files 9](#_Toc484760976)

[1.10 Data Search 9](#_Toc484760977)

[1.11 Metadata 10](#_Toc484760978)

[1.12 Generating Reports 10](#_Toc484760979)

[1.13 Role and Authorization 10](#_Toc484760980)

[1.14 Notifications 12](#_Toc484760981)

[1.15 Data Integrity 13](#_Toc484760982)

[1.16 Storage and Supported Data Types 13](#_Toc484760983)

[Business Rules for Functional Requirements 13](#_Toc484760984)

[Non-Functional Requirements 13](#_Toc484760985)

[1.17 Hardware Requirements 13](#_Toc484760986)

[1.18 Software Requirements 13](#_Toc484760987)

[1.19 Performance Requirements 16](#_Toc484760988)

[1.20 Reliability Requirements 16](#_Toc484760989)

[1.21 Supportability Requirements 16](#_Toc484760990)

[1.22 User Documentation Requirements 17](#_Toc484760991)

[1.23 Interface Requirements 17](#_Toc484760992)

[1.24 Security and Privacy Requirements 17](#_Toc484760993)

[1.25 Compliance and Standards Requirements 17](#_Toc484760994)

[Appendix A: Requirements Definition Approval 18](#_Toc484760995)

[APPENDIX B: REFERENCES 19](#_Toc484760996)

# Introduction

## Purpose of the Document

This Requirements Definition document describes the high level functional, non-functional, and technical requirements as gathered from the CBIIT, CCR and DCEG customers for the prototyping effort to develop a HPC foundation for the NCI HPC enterprise. The Requirements Definition document is created during the Requirements Analysis Phase of the project and has been revised/enhanced along the agile development/implementation of this effort. The document describes the hardware and software requirements and constraints as well as any assumptions and dependencies concerning the system.

This document can be used as a living document for developing both technical or non-technical project specifications. It can be used to assess, validate and update pre-existing requirements.

# Business Requirements Overview

The HPC Data Management Environment (DME) prototyping effort shall work under the guidance of NCI CBIIT Government Sponsor, LBR Director HPC strategy and TPM and establish a HPC foundation for the NCI HPC enterprise. The HPC efforts envision a data management system for NCI which can provide a secure and efficient infrastructure for physical storage, easy query and retrieval of data sets, staging data for analysis and also efficient recomputing of results while dealing with huge high through put data.

## Assumptions / Constraints

The developed or proposed system will be hosted within NCI/Leidos premises.

## System Scope

The main scope of this effort is to conduct requirements gathering sessions with NCI Center for Cancer Research (CCR) and Division of Cancer Epidemiology and Genetics (DCEG) customers, develop/prioritize business use case scenarios through Leidos Biomedical Research (LBR) technical project manager (TPM), prototype and implement technical APIs to decouple HPC data storage from usage of the data via an iterative agile development methodology. The developed APIs, associated dataset object repository, metadata and registration catalog database shall empower CBIIT in unifying its IT services, infrastructure and high performance computing architecture throughout NCI to support NCI enterprise business needs.

Some high level objectives of this initiative are:

1. Establish GridFTP services to support general transfer of large datasets without requiring physical mounting.
2. Establish a pilot dataset registration system to associate a label with a given managed dataset. The System will capture extensible metadata including security and access requirements for the managed dataset. Metadata includes but not limited to: How the dataset was generated, when it was generated, where it was generated, present access method and information to obtain a full copy of the original dataset. System will also be flexible to support either a) export of metadata to a future system; b) development of service APIs to support interrogation by secondary systems; or c) both (a) and (b).
3. Establish easy-to-use methods for providing annotation information and creating reports of managed datasets, including utilization of a modest controlled vocabulary related to high-use criteria for searching datasets.
4. Establish a high-reliability storage model for underlying datasets included into the data management environment (DME).
5. Obtain utilization statistics for managed datasets.

# Functional Requirements

## User Login and Logout

### The HPC DME shall follow NIH AD user accounts policies.

### The HPC DME shall allow any user with an NIH account to log into the HPC DME using the NIH account credentials on the UI or via API.

### The HPC DME shall validate passwords for the user accounts through NIH AD on the UI or via API.

### The HPC DME shall prompt the user to try logging in with valid credentials after an unsuccessful login.

### The HPC DME shall allow users to log out of the HPC DME system on the UI or via API. On UI, redirect the user to login page.

### The HPC DME shall automatically logout a user during session inactivity on the UI or via API.

## User Home page/Dashboard

### The HPC DME shall direct the user to user dashboard after login where user can view a list of:

#### All collections (projects, directories or datasets) registered by the user

#### All collections (projects, directories and datasets) s/he is authorized to view

### The HPC DME shall allow user subscriptions to events like data transfer, or metadata update tasks performed.

### The HPC DME shall provide capabilities for performing tasks such as registering new collections, data files, searching collections, data files, transferring collections/data files, adding Globus end points, creating groups, adding users and setting permissions for users.

### The HPC DME shall allow data owners to manage sharing data/metadata with other users or groups.

## Collection Registration

### The HPC DME shall facilitate registering of independent collections with the HPC DME via a UI or an API.

### The HPC DME shall generate a unique HPC ID for each of the independent collection registered.

### The HPC DME shall facilitate adding one or more new files to the collections already registered with HPC via UI or API.

### The HPC DME shall mandate submission of ‘mandatory’ metadata, if specified in the policy file, at the time of collection registration via a UI or an API.

### The HPC DME shall provide capabilities to enter user-defined metadata while registering a collection via a UI or an API.

### The HPC DME shall provide flexibility to extend the MIOMI biomedical model in assisting different labs or groups annotating their data.

## ****File Registration****

### The HPC DME shall facilitate registering of File (not associated to any collection) with the HPC DME via an API.

### The HPC DME shall generate a unique HPC ID for each of the Files registered.

### The HPC DME shall facilitate associating one or more registered files to registered collections already registered with HPC DME.

### The HPC DME shall mandate submission of ‘mandatory’ metadata, if specified in the policy file, while registering a file via an API.

### The HPC DME shall provide capabilities to enter user-defined metadata while registering a file via an API.

## Data Transfer

### The HPC DME shall integrate with and use Globus platform to perform any kind of data transfer tasks.

### The HPC DME shall provide capabilities to authorized users to transfer one or more datasets from any authorized source Globus end point to the HPC target Globus end point, via API.

### The HPC DME shall provide capabilities to authorized users to transfer one or more datasets from HPC source Globus end point to any authorized target Globus end point, via API.

### The transfer of datasets to and from the HPC Globus end point shall be asynchronous via API.

### The HPC DME shall provide a status of all the data transfer processes started by the HPC user via the API.

### The HPC DME shall provide synchronous data transfer on a single file

### The HPC DME shall allow end users to retrieve Globus transfer status information: endpoint names, source, target path, status of the data transfer via API, for each transfer.

## Bulk Transfer of datasets/ files

### The HPC DME shall facilitate bulk transfer of collections, for example 500 NGS files.

### The HPC DME shall facilitate bulk upload of metadata for collections already uploaded and registered with HPC DME archive.

### The HPC DME shall facilitate bulk upload of collections or associated metadata using command-line functionality.

## Data Search

### The HPC DME shall provide a search facility to the HPC users to search for data objects via APIs, based on:

* Metadataprovided by the user at the time of collection registration
  + Collection or File name
  + Associated metadata
* System generated metadata
  + Collection ID
  + Registration Date (Time window based on registration dates)
  + File Transfer Status
* User name/ID

### The HPC DME shall facilitate users to perform complex searches using more than one search criteria. These criteria may have combinations of metadata attributes from different levels of collections or data objects themselves. The HPC DME user needs to search desired data object (s) registered with HPC DME, using any available metadata associated with the object itself or any of its parent collections at any levels via APIs. This is based on the iRODS iCAT schema implementation with supported collection hierarchies.

### The HPC DME shall present data objects or collections registered by the user and those the user is authorized to view, if the user performs a search without entering any search criteria, via APIs.

## Metadata

### The HPC DME shall store all the metadata provided by the HPC DME users, for each collections registered with HPC.

### The HPC DME shall provide API capabilities to the user to enter the metadata for each collection or file registered with HPC.

### The HPC DME shall provide capabilities to add new user defined metadata variables for each collection or file via UI or API, both at the time of registration and even later.

### The HPC DME shall provide capabilities for related users or groups of users to be able to modify/update metadata associated to a collection or a file via UI or API.

### System Generated Metadata

#### The HPC DME shall generate and save system related metadata for each collection or data file after transfer to HPC DME via a UI or an API.

## Generating Reports

### The HPC DME shall generate following reports periodically or as needed for all users and/or groups registered with HPC.

* Access and Utilization report by User
* Access and Utilization report by Group
* Metadata coverage of data objects

### These reports generated capture information on storage access and utilization by each user or group/s and the metadata coverage of data objects.

## Role and Authorization

### The HPC DME shall have the following roles for HPC users:

* System Administrator
* Group Administrator (Super User)
* User

### The HPC DME shall provide capability to grant the following authorizations to the HPC roles mentioned above:

* Create data object
* View data object
* Share data object
* Stop Sharing data object
* Update data object
* Create Metadata
* Update Metadata
* Delete Metadata
* Delete data object

### Creating Groups or Adding users

#### The HPC DME shall allow System Administrator or Group Administrator to create groups and add users to the groups on HPC DME via a UI or an API.

### Data Sharing

#### The HPC DME shall allow authorized user to share one or more of their collections with other users by: Granting other user required read, write, own or NONE permission directly. Or adding the users to a user group/s in the HPC DME and setting up group permissions on the collections, via API

#### The HPC DME shall allow authorized user to share one or more of their collections with other users by: Granting other user required read, write, own or NONE permission directly. Or adding the users to a user group/s in the HPC DME and setting up group permissions on the collections, via UI

#### The HPC DME shall allow authorized user to share one or more of their data objects with other users by: Granting other user required read, write, own or NONE permission directly. Or adding the users to a user group/s in the HPC DME and setting up group permissions on the data objects, via API

#### The HPC DME shall allow authorized user to share one or more of their data objects with other users by: Granting other user required read, write, own or NONE permission directly. Or adding the users to a user group/s in the HPC DME and setting up group permissions on the data objects, via UI

### Metadata Sharing

The authorized HPC DME user wants to share his/her data or metadata with

collaborators/Analysts or PI for research or analysis purposes but only for certain

metadata only. The user provides the collaborators/Analysts with

* Metadata related to the data object itself or it parent collections, and
* Permission to download the datasets.

The basic requirements on authorizations are:

#### Whoever investigator is assigned as the PI on a CSAS Project, will have read access to all metadata associated with the project and all its children collection and data files (objects), i.e., the PI will either have read access to all metadata or nothing at all (for which they are not the assigned PI in the latter case).

#### Current support scenarios are that we will either grant all metadata attributes access or deny all to a user or group on a collection or data file

#### There will be a need to support authorizations (for either group or individual user) by inheritance, i.e., a user/group with metadata read access on a CSAS project level, by default, will have read access to all children and datafiles’ metadata under the same project.

## Notifications

### The HPC DME shall notify the HPC DME system administrator group DL list group ([HPC\_DME\_Admin@nih.gov](mailto:HPC_DME_Admin@nih.gov)) only when a Globus asynchronous data transfer (either upload or down load) fails due to whatever reasons.

### The HPC DME shall notify the HPC project owner or authorized user (with a minimum read access permission granted) when an upload is completed successfully and if the related user subscribes to such event. A user will not get such notifications if s/he has not made the subscription.

### The HPC DME shall notify the HPC project owner or authorized user when an upload is not successful and if the related user subscribes to such event. A user will not get such notifications if s/he has not made the subscription.

### The HPC DME shall notify the HPC project owner or authorized user when a download is completed successfully and if the related user subscribes to such event. A user will not get such notifications if s/he has not made the subscription.

### The HPC DME shall notify the HPC project owner or authorized user when a download is not successful and if the related user subscribes to such event. A user will not get such notifications if s/he has not made the subscription.

## Data Integrity

#### The HPC DME shall perform an automatic data integrity test every time a data file is moved or copied.

#### The DME shall use checksums to ensure data integrity after each copy or move.

## Storage and Supported Data Types

### The HPC DME shall facilitate data storage of all the supported data types uploaded by HPC users. The data could be generated from high through put technologies such as Next Generation Sequencing, Imaging, Proteomics etc.

### The HPC DME shall provide capability to add new scientific data types.

# Business Rules for Functional Requirements

The different criteria and conditions associated with the functional requirements are listed here as the business rules.

### Collection Names shall be unique under the same parent collection.

### File Names shall be unique under the same collection.

### A data object may be linked to one or more collections and vice versa.

### Mandatory metadata attributes shall be entered into the organization’s HPC Policy file and the ‘mandatory’ clause set to true at the time of registration. HPC DME shall mandate the submission of those set attributes at the time of registration.

# Non-Functional Requirements

## Hardware Requirements

[Describe hardware requirements and any related processes. Include a detailed description of specific hardware requirements and associate them to specific project functionality/deliverables. Include information such as type of hardware, brand name, specifications, size, security, etc. Assign a unique ID number to each requirement.]

## Software Requirements

HPC System comprises of HPC Server and HPC Web modules. HPC Server module shall be responsible for all the Server-side sub-modules required to support functional requirements through APIs. HPC Web module shall be responsible to support all user interface requirements using Server-side APIs.

### HPC Server

#### HPC server modules should follow SOA methodology to enable loosely coupled, highly customizable, autonomous, composable and discoverable APIs supporting user requirements.

#### HPC Server shall provide Data Transfer API to transfer data from any location to HPC data store. Data transfer API shall take file/directory location, and transfer dataset from origin to destination. Origin and destination would need to be HPC data endpoints, local or network shares in order to enable to the transfer.

| ***Req. ID*** | ***Requirement*** |
| --- | --- |
| DT-1 | Data Transfer API shall support both uploading large datasets to the HPC data store and downloading large datasets from HPC data store to another HPC data endpoint. |
| DT-2 | Data Transfer API shall provide a uniform way of accessing the data to enable high-speed, reliable, and secure data transfer of large datasets. |
| DT-3 | Data Transfer API shall integrate with Globus online security to enable Globus online users to access the API. |
| DT-4 | Data Transfer API shall provide a fault tolerant implementation of FTP, to handle network unavailability and server problems. Transfers should also be automatically restarted if a problem occurs. Data Transfer API shall return result of the data transfer request submission. The result should include any validation messages in case of failed state. |
| DT-5 | Data Transfer API should be an asynchronous process where the API accepts requests and process them in the background. Data transfer API shall follow Connector (Adapter) pattern to support pluggable connectors to do actual transfer. One example would be Transfer API should provide Globus Connect Server adapter to perform transfers between Globus endpoints. Transfer API shall have configuration to support replacing or adding adapters. |
| DT-6 | Data Transfer API shall provide status of an asynchronous transfer request. |

#### HPC Server shall provide Data registration API to register data with HPC data store. Registration API is a composite API that works with Metadata API and Data Transfer API to store the dataset in the HPC data store and its associated metadata in the HPC metadata store with reference to the data.

| ***Req. ID*** | ***Requirement*** |
| --- | --- |
| DR-1 | Data registration API shall support registering a collection into HPC Data store. The registration should capture information about the collection and its metadata. |
| DR-2 | Data registration API shall support validating collection and metadata based on validation rules set by Validation API. Data registration API shall return any validation messages as the response. |
| DR-3 | Data registration API shall store metadata in HPC metadata store and data objects into HPC archive store. Data registration API shall create a reference between data object and its metadata. |

#### HPC Server shall provide Metadata API to management metadata life cycle that includes create, update and delete. Metadata API shall need to work with Security API to enforce authorization on the metadata management.

| ***Req. ID*** | ***Requirement*** |
| --- | --- |
| MD-1 | Metadata API shall support adding various formats of metadata into HPC metadata store. |
| MD-2 | Metadata API shall support validating metadata based on validation rules set by Validation API. Metadata API shall return any validation messages as the response. |
| MD-3 | Metadata API shall work with Security API to enforce security at document and attribute level. Metadata API shall support Read, Write and Delete operations on metadata. |

#### HPC Server shall provide Search API to search dataset based on metadata.

#### HPC Server shall provide Notification API to generate push-pull notifications on metadata and dataset lifecycle changes.

#### HPC Server shall provide Security API to secure all above mentioned APIs through authentication and authorization.

#### HPC Server shall provide Validation API to enable auditing on dataset and metadata management.

### HPC DME Web

#### HPC DME Web shall use open source technologies and provide customizable solution supporting user interface requirements.

#### HPC DME Web shall provide collection registration interface to register collection and upload data from a location to HPC data store. Data upload interface shall work Data registration API to register and transfer data from origin to destination. Origin and destination would need to be HPC data endpoints in order to enable to the transfer for asynchronous transfer, or a local or network share/directory for synchronous transfer.

#### HPC DME shall incorporate “Not Specified” as a default value for all the user input variables where the user does not enter any data.

## Performance Requirements

[Describe performance requirements and any related processes. Include a detailed description of specific performance requirements and associate them to specific project functionality/deliverables. Include information such as system capacity, cycle time, speed per transaction, test requirements, minimum bug counts, speed, reliability, utilization etc. Assign a unique ID number to each requirement.]

## Reliability Requirements

[Describe all of the technical requirements that affect availability such as hours of operation, level of availability required, down-time impact, support availability, accuracy, etc. Assign a unique ID number to each requirement.]

## Supportability Requirements

#### The HPC DME provides the flexibility allowing a DOC or lab using its own separate object vault in the backend storage

#### Each DOC or lab will have its own group administrator owning the data and metadata, and be responsible on how data hierarchic and data life cycle (or policies) will be maintained

## User Documentation Requirements

### The HPC DME project shall provide all the following documentation:

* HPC DME Design/Integration Document (SDID)
* HPC DME Requirements Definition
* HPC DME DR and Monitoring Plan
* HPC DME Server API specification
* HPC DME System Administration Guide
* HPC DME User Guide
* HPC DME General Training

## Interface Requirements

[Describe all of the user interface (such as user navigation, presentation of application and associated functionality, screen location of interface elements, data display and manipulation, etc), system interface, and technical (hardware and software) requirements that affect interfaces such as look-and-feel, protocol management, scheduling, directory services, broadcasts, message types, error and buffer management, security, etc. Assign a unique ID number to each requirement.]

## Security and Privacy Requirements

[Summarize and make reference to Privacy Impact Assessment produced during Planning phase and its impact on security requirements. Provide justifications for why a specific privacy item is needed. Provide Security Categorization if available.

Describe all of the technical requirements that affect security such as security audits, cryptography, user data, system identification/authentication, resource utilization, facility access times, etc. Assign a unique ID number to each requirement. ]

## Compliance and Standards Requirements

*[Describe the existing compliance environment as it affects project requirements, and the standards the system development must follow. Include an overview of the compliance or standards requirements necessary to achieve the project’s objectives. List all that are applicable to the project. Assign a unique ID number to each requirement.]*

### Section 508 Compliance

*[This section applies to the systems that are required to be Section 508 compliant. Describe if and how the Section 508 of the Rehabilitation Act affects the system; cite the technical standards it must meet. Assign a unique ID number to each requirement.]*

Appendix A: Requirements Definition Approval

The undersigned acknowledge that they have reviewed the ***<Project Name>* Requirements Definition** 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.

[List the individuals whose signatures are desired. Examples of such individuals are Business Owner, Project Manager (if identified), and any appropriate stakeholders. Add additional lines for signature as necessary.]

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

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

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

APPENDIX B: REFERENCES

[Insert the name, version number, description, and physical location of any documents referenced in this document. Add rows to the table as necessary.]

The following table summarizes the documents referenced in this document.

|  |  |  |
| --- | --- | --- |
| **Document Name** | **Description** | **Location** |
| Primary Metadata Variables.docx | Lists all the required metadata variables. | https://ncisvn.nci.nih.gov/svn/HPC\_Data\_Management/branches/hpc-prototype-dev/doc/requirements/ |
| Secondary Metadata Variables.docx | Lists all the domain specific metadata variables. | https://ncisvn.nci.nih.gov/svn/HPC\_Data\_Management/branches/hpc-prototype-dev/doc/requirements/ |