Getting Started with the Globus Replica Catalog

Introduction

This document describes the Globus Replica Catalog, which is used to register and manage complete and partial copies of data sets. Replica management is an important issue for a number of scientific applications. For example, consider a data set that contains one petabyte of experimental results for a particle physics application. While the complete data set may exist in one or possibly several physical locations, it is likely that many universities, research laboratories or individual researchers will have insufficient storage to hold a complete copy. Instead, they will store copies of the most relevant portions of the data set on local storage for faster access.

Services provided by a replica catalog include:

- Registering a list of files as a ***logical collection***
- Registering the physical ***location*** of a complete or partial replica of a logical collection
- Registering information about a particular ***logical file*** in a logical collection
- Modifying the contents of registered entries in the catalog
- Responding to queries of the catalog, such as:
    - Find all physical locations for a set of logical files in a logical collection
    - List all the descriptie attributes associated with a registered logical collection, location or logical file.

This document begins by describing the Replica Catalog in detail, including a discussion of the pre-defined attributes that can be used to describe logical collections, locations and logical files that are registered in the catalog. Next, we present the command-line functions that manipulate replica catalog entries. The document concludes with an example data replication scenario and shows how command-line functions can be used to create a replica catalog for this scenario.

Replica Catalog Entries and their Attributes

As mentioned above, the purpose of the replica catalog is to provide mappings between logical names for files or collections and one or more copies of the objects on physical storage systems. The catalog registers three types of **entries**: logical collections, locations and logical files. Users may associate descriptive **attribute:value** pairs with each entry registered in the catalog.

In this section, we describe each entry type in the replica catalog as well as the required and optional attributes associated with these entries.

Logical Collections

A logical collection is a user-defined group of files. We expect that users will find it convenient and intuitive to register and manipulate groups of files as a collection, rather than requiring that every file be registered and manipulated individually. Aggregating files should reduce both the number of entries in the catalog and the number of catalog manipulation operations required to manage replicas.

When registering a group of files as a logical collection, the user may associate five optional attribute:value pairs with the collection:

- The optional **filename** attribute allows the user to specify the names of logical files that belong to the collection. This attribute accepts case-sensitive string values. We currently require that each filename registered in the Replica Catalog be associated with a single logical collection, although we plan to relax this requirement in the future.

- The optional **creatorsname** and **modifiersname** attributes are used to identify the creator and most recent modifier of the logical collection. These attributes accept LDAP distinguished name (DN) values in our current implementation.

- The optional **createtimestamp** and **modifytimestamp** attributes allow the user to associate timestamps with the logical collection indicating when the entry was created or most recently modified. These attributes accept numeric values.

Note that logical collection entries contain no attributes that provide information required to map from logical collection and file names to physical locations. Logical collection entries simply provide a mechanism for grouping logical files together. The replica catalog places no restrictions on the list of files specified as a logical collection. The catalog does not even require that any of the logical file names registered as part of a collection exist on a physical storage system.

It is also notable that the logical collection entries in the replica catalog do not contain any attributes that describe the contents of the logical collection or its constituent files. Such descriptive information or ***metadata*** is assumed to exist in a separate catalog. These attribute are deliberately excluded from the replica catalog, since its sole purpose is to provide mappings between logical names for files and collections and physical storage locations.

Locations

Location entries in the replica catalog contain all the information required for mapping a logical collection to a particular physical instance of that collection. The location entry may register information about the physical storage system, such as the hostname, port and protocol. In addition, it contains all information needed to construct a URL that can be used to access particular files in the collection on the corresponding storage system.

Each location entry represents a complete or partial copy of a logical collection on a storage system. One location entry corresponds to exactly one physical storage system location. The location entry explicitly lists all files from the logical collection that are stored on the specified physical storage system.

Location entries in the replica catalog have one required attribute and nine optional attributes:

- One **uc** attribute is *required* for each location entry. The uc or URL constructor attribute provides the mapping mechanism from logical names to physical names. The attribute specifies a grammar that the replica catalog uses to generate a complete URL, which is required by transport protocols that read or write individual files. Typically, the URLconstructor contains the protocol, hostname, optional port number, and file system directory path of the storage system that holds the replica. In the current implementation, the URL constructor is a string that provides a prefix for the URL, to which the logical filenames are appended to generate a complete URL. In the future, we will provide a more general grammar for mapping from arbitrary logical file names to possibly different physical file names. The attribute accepts case-insensitive string values.

- The optional **filename** attribute is used to specify that a particular logical file in the collection is present at this physical storage location. This attribute accepts case-sensitive string values. Each location entry must contain one filename attribute for each file stored at this physical location or *replica*. Using filename attributes thus allows us to specify either complete or partial replicas using the same location entry type.

- The optional **hostname** attribute specifies the host machine for the physical storage system, such as myhost.isi.edu. This attribute accepts case-insensitive string values.

- The optional **protocol** attribute specifies the access protocol required by the storage system. Examples include ftp, http, gridftp and dpss. The attribute accepts case-insensitive string values.

- The optional **port** attribute specifies the port number required to access the storage system using the specified access protocol. The attribute accepts numeric values.

- The optional **path** attribute specifies the file system path information for the storage system directory where the replica is stored. The attribute accepts case-insensitive string values

- The optional **creatorsname** and **modifiersname** attributes are used to identify the creator and most recent modifier of the location entry. These attributes accept LDAP distinguished name (DN) values in our current implementation.

- The optional **createtimestamp** and **modifytimestamp** attributes allow the user to associate timestamps with the location indicating when the entry was created or most recently modified. These attributes accept numeric values.

Each logical collection entry may have an arbitrary number of associated location entries, each of which contains a (possibly overlapping) subset of the files in the collection. Using multiple location entries, users can easily register logical collections that span multiple physical storage systems.

Logical Files

Despite the benefits of registering and manipulating collections of files using logical collection and location objects, users and applications may also want to characterize individual files. For this purpose, the replica catalog includes optional entries that describe individual ***logical files***. Logical files are entities with globally unique names that may have one or more physical instances. The catalog may optionally contain one logical file entry in the replica catalog for each logical file in a collection.

Logical file entires have five optional attributes:

- The optional **size** attribute specifies the size of the logical file being registered in the catalog. This attribute accepts a numeric value in units of bytes.
- The optional **creatorsname** and **modifiersname** attributes are used to identify the creator and most recent modifier of the logical file entry. These attributes accept LDAP distinguished name (DN) values in our current implementation.
- The optional **createtimestamp** and **modifytimestamp** attributes allow the user to associate timestamps with the logical file entry indicating when the entry was created or most recently modified. These attributes accept numeric values.

Like logical collection entries, logical file entries contain no attributes that describe the contents of logical files (metadata) or the physical location of files on storage systems.

The Command Line Tool for Manipulating the Replica Catalog

In this section, we present the functions provided by the command-line tool for manipulating replica catalog objects. All operations supported by the command line tool use the following format:

globus-replica-catalog HOST OBJECT ACTION

Specifying the HOST

The HOST argument for the command specifies the location of a particular logical collection in the replica catalog. We currently implement the replica catalog as an LDAP directory specified by a URL. A typical URL for a catalog is of the format:

ldap://host[:port]/dn

This URL specifies the hostname of the LDAP directory server as well as the port number, if it is non-standard. In addition, it specifies the distinguished name (DN) of the logical collection in a particular replica catalog.

In addition to specifying the URL for the replica catalog, the HOST section of the command also includes options for specifying the directory manager and a file name that contains a password for the LDAP directory. The directory manager is allowed to modify the contents of an LDAP directory, if the password is specified correctly. This provides simple password-based authentication to control modifications of the replica catalog. The Globus project will soon replace this password-based authentication scheme with PKI certificate-based authentication on LDAP directories.

Below, we show the command line tool's documentation for the input parameters related to the HOST portion of the command:

```
          HOST
           -host <collection url>
             connect to a host of the given LDAP URL of the form
             ldap://host[:port]/dn which includes the DN of a collection, can
             be set with the GLOBUS_REPLICA_CATALOG_HOST environment
variable
           [ -manager <manager DN> ]
             DN to be used during authentication, can be set with the
             GLOBUS_REPLICA_CATALOG_MANAGER environment variable
           [ -password <input file> ]
             password to be used during authentication
```

Specifying the OBJECT

The next part of the command is the OBJECT portion, which indicates what entry in the replica catalog the command will act upon. If the user specifies the "-collection" option, then no additional argument is required; the operation affects the collection entry specified in the "-host" argument. The other objects that can be operated upon are a "-location" entry or a "-logicalfile" entry associated with the specified logical collection. For each of these command options, the user must supply the name of the location or logical file entry.

The command line tool's documentation for the input parameters related to the OBJECT portion of the command are shown below:

```
          OBJECT
           -collection
             act on the collection given in the DN of the LDAP URL
           -location <location name>
             act on the given location name
           -logicalfile <logical file name>
             act on the given logicalfile name
```

Specifying the ACTION

The final portion of the command specifies the ACTION that the command will perform on the specified entry. These operations fall into four general categories: create and delete entire entries; manipulate (add, list or delete) attributes of an entry; manipulate filename attributes of a logical collection or location entry; list or search for specified entries.

Create and Delete Operations

The arguments to the "-create" parameter vary depending on the entry specified in the OBJECT portion of the command. To create a new collection entry, the user must supply an input file that contains a list of filenames to be registered as members of the new collection. Each logical file listed in the input file will be added as a filename attribute of the logical collection entry.

To create a new location entry, the user must provide the URL of the physical location where the replica of the collection is stored. During the object creation, the command line tool parses this URL to generate the uc (URL constructor), hostname, protocol, port and path attributes for the new location entry. The create command for a new location object optionally includes an input file that specifies a list of filenames stored in the location storage system. This file may contain a full or partial list of the files in the logical collection. Each file listed in the input file is added as a filename attribute of the location entry.

Last, to create a new logical file entry, the user specifies the value to be assigned to the size attribute in the logical file entry.

To delete an entry in the catalog, the user simply specifies the "-delete" option in the ACTION section of the command. Note that a delete operation for a logical collection will fail if the collection if any entries exist for logical files or locations associated with that collection.

The command line documentation associated with these commands follows:

```
ACTION
 -create [ <input file> ] | -create <location url> [ <input file> ]
       | -create <size>
    create a collection with filenames in the given input file,
    a location with the given location url and optional set of
    filenames in the given input file, or a logicalfile with the
    given size
  -delete
    delete a collection, location, or logical file
```

Manipulating Arbitrary Attributes

For a replica catalog entry specified in the HOST and OBJECT portions of the command, the user may select the following actions: "-add-attribute", "-delete attribute" or "list-attributes". The user is allowed to manipulate the pre-defined attribute for each entry. When adding an attribute to an entry, the user must specify both the attribute name and its value. When deleting an attribute, the user must specify the attribute name and optionally may specify a specific value to be deleted. The user mayh also list attributes of an entry. If particular attribute names are specified, the operation returns only those attributes. If no attribute names are specified, the operation returns all attributes associated with a replica catalog entry.

The command line documentation associated with these commands follows:

```
ACTION
 -add-attribute <attribute name> <attribute value>
    add the given attribute name and value to a collection
    location, or logical file
  -delete-attribute <attribute name> [ <attribute value> ]
    delete an attribute or optionally the specific value of an
    attribute from a collection, location, or logical file
  -list-attributes [ <attribute names> ]
    list attributes of a collection, location or logical file,
    returning all attributes or optionally only the given attributes
```

Manipulating Filename Attributes of Logical Collections and Locations

Logical file names are of central importance in a replica catalog, since the catalog's main purpose is to map from logical file names to physical file names. Because of the importance of these attributes, we have defined several functions that operate only on filename attributes found in logical collection and location entries.

The "-add-filenames" operation requires an input file containing a list of filenames to be added as attributes to a logical collection or location entries. One question that arises is the behavior of the function if some of the filename attributes specified in the input file are already present in the collection or location entry. If the user specifies the "-add-anyway" option, then this command will add the filenames that are not present in the collection or location. If the "-add-anyway" option is not specified, then the operation fails if any of the filenames specified in the input file match existing filename attributes of the location or logical collection entry.

The "-delete-filenames" operation requires an input file containing a list of filenames to be deleted from a logical collection or location entry. If the input file contains the names of any filenames that are not present as attributes of the specified entry, then the operation fails and no deletions are performed. For a logical collection entry, the delete-filenames operation also fails if there are any location or logical file entries associated with the collection that contain the specified filenames.

The "list-filenames" operation lists all filename attributes of a logical collection or location entry.

The command line documentation associated with these commands follows:

```
ACTION
  -add-filenames <input file> [ -add-anyway ]
    add filenames in the given input file to a collection or
    location, optionally add the filenames that aren't there if some
    of the filenames are already there anyway
  -delete-filenames <input file>
```

delete filenames in the given input file from a collection
or location
-list-filenames
list filenames in a collection or location

List and Search Operations on Entries

The final category of operations supported by the command line tool are list and search operations on entries in the replica catalog.

The "-list-locations" operation lists all location entries associated with the specified logical collection. The user may optionally specify a list of attribute names. If attribute names are specified, then only those attribute names and their values are returned for each location listed by the operation. If no attribute names are specified, then the operation returns all attributes of the location entries listed by the operation.

The "-list-logicalfiles" operation lists the attributes of all logical files entries associated with the specified logical collection. The user may optionally restrict the attributes returned by the operation by specifying attribute names.

One of the most important commands in the replica catalog command line tool is the "-find-locations" command. This search command takes as input a list of logical filenames. The command searches for location entries that contain filename attributes for ALL the specified filenames, indicating that the physical storage system associated with the location entry contains a copy of each of these files. The command returns URLs that specify the physical locations of the matching files. This command is used by high level tools that locate and select among replicas of a complete or partial logical collection. The user may optionally restrict the attributes returned by the search operation by specifying attribute names.

The "-list-collections" operation is unusual, since it requires no OBJECT portion of the command. Using the HOST portion of the command, this operation returns the names of all logical collections in the catalog.

The command line documentation associated with these commands follows:

ACTION
-list-locations [<attribute names>]
list locations in a collection, optionally restrict the
attributes returned to the attribute names given
-list-logicalfiles [<attribute names>]
list logical files in a collection, optionally restrict the
attributes returned to the attribute names given
-find-locations <input file> [-attributes <attribute names>]
find locations in a collection that have filenames in the given
input file, optionally restrict the attributes returned to the
attribute names given
-list-collections
list collections in a replica catalog, the DN in the given host

LDAP URL is the base from which to search, no object need be specified

Command Line Tool Examples

We provide some simple examples for creating, listing and deleting objects using the command line tool.

Register a collection

```
globus-replica-catalog \
  -host “ldap://myldap.mcs.anl.gov:389/lc=newLogicalCollection,\
    rc=myCatalog,o=Argonne National Laboratory,o=Globus, c=US”\
  -manager “cn=Diretory Manager,o=Argonne National Laboratory,\
    o=Globus, c=US” –password pwfile \
  -collection –create listOfCollectionFilenames
```

Register a location

```
globus-replica-catalog \
  -host “ldap://myldap.mcs.anl.gov:389/lc=newLogicalCollection,\
    rc=myCatalog,o=Argonne National Laboratory,o=Globus, c=US”\
  -manager “cn=Diretory Manager,o=Argonne National Laboratory,\
    o=Globus, c=US” –password pwfile \
  -location “location1” –create “gridftp://dc-n1.isi.edu/nfs/mydir/mypath/” \
    listOfLocationFilenames
```

List all logical file names in a collection

```
globus-replica-catalog \
  -host “ldap://myldap.mcs.anl.gov:389/lc=newLogicalCollection,\
    rc=myCatalog,o=Argonne National Laboratory,o=Globus, c=US”\
  -manager “cn=Diretory Manager,o=Argonne National Laboratory,\
    o=Globus, c=US” –password pwfile \
  -collection –list-filenames
```

List all attributes of a location

```
globus-replica-catalog \
  -host “ldap://myldap.mcs.anl.gov:389/lc=newLogicalCollection,\
    rc=myCatalog,o=Argonne National Laboratory,o=Globus, c=US”\
  -manager “cn=Diretory Manager,o=Argonne National Laboratory,\
    o=Globus, c=US” –password pwfile \
  -location “location1” –list-attributes
```

Delete a location

```
globus-replica-catalog \
  -host “ldap://myldap.mcs.anl.gov:389/lc=newLogicalCollection,\
    rc=myCatalog,o=Argonne National Laboratory,o=Globus, c=US”\
  -manager “cn=Diretory Manager,o=Argonne National Laboratory,\
    o=Globus, c=US” –password pwfile \
  -location “location1” –delete
```

Delete a collection

```
globus-replica-catalog \
  -host “ldap://myldap.mcs.anl.gov:389/lc=newLogicalCollection,\
    rc=myCatalog,o=Argonne National Laboratory,o=Globus, c=US”\
  -manager “cn=Diretory Manager,o=Argonne National Laboratory,\
    o=Globus, c=US” –password pwfile \
  -collection –delete
```

An Example Replication Scenario

Replication Scenario

A user wants to register partial replicas of a logical collection at two sites, A and B. The complete collection contains a total of five files with the following file sizes:

- file 1: 100MB
- file 2: 200MB
- file 3: 300MB
- file 4: 400MB
- file 5: 500MB

The partial collection stored at Site A contains:

- file1
- file2
- file3
- file4

The partial replica stored at Site B contains:

- file 2
- file 3
- file 5

In addition, the user wants to search for all physical locations of a file (for example, file 2). The user also wants to query the replica catalog for size information on the selected file. This information is used by higher-level performance estimation tools to select among replicas based on predicted performance of data transfers.

Replica catalog implementation:

This replication scenario requires that the user register the following entries in the replica catalog:

- One logical collection entry containing all five logical filenames as attributes
- One location entry for sites A and B specifying the files stored at each physical location
- Five logical file entries specifying the size of each logical file

For the collection creation operation, the user must specify the name of the new collection as well as an input file that lists all the logical file names that will be associated with this collection. For our scenario, this input file (called, for example, **listCollectionNamesFile**) contains the following names:

file1
file2
file3
file4
file5

For the two sites A and B, the user must register two location entries in the replica catalog. For each location entry, the user must specify the required uc or URLconstructor attribute that describes the mapping from logical names to physical storage locations. This consists of the protocol, hostname, port and path information for each site. In addition, each location entry must contain a filename attribute for each file stored on the storage system.

For the location entry corresponding to site A, the user might specify the following value for the uc attribute:

uc: gridftp://Ahost.isi.edu:2222/nfs/path/on/A/

The location entry command also requires a file (e.g., **listANamesFile**) that contains a list of the files stored at location A:

filename: file1
filename: file2

filename: file3
filename: file4

Similarly, the location entry corresponding to site B must also specify a value for the uc attribute:

Uc: ftp://Bhost.mcs.anl.gov:7777/nfs/path/on/B/

The input file (e.g., **listBNamesFile**) containing file names stored at location B is shown below:

filename: file2
filename: file3
filename: file5

During the creation operations for the five logical file entries, the user must supply values for the size attribute for each entry in units of bytes. The size values for the five files in the logical collection are:

file1:	**104857600**
file2:	**209715200**
file3:	**314572800**
file4:	**419430400**
file5:	**524288000**

After all these entries have been created, the user can search entries in the catalog. For example, the user can request the names of all locations that contain file2 or all locations that contain file4 AND file5. To perform the latter search operation, the user must supply a file (e.g., **namesToSearchFile**) that contains the names of all requested files:

file4
file5

The user can also make a separate query to request the size of a logical file entry. Using location and file size information, higher level tools might perform replica selection based on performance estimates provided by tools such information services as the Network Weather Service, which provides network performance information, and the Grid Information Service, which provide storage system performance estimates.

Implementing this Scenario with the Command Line Tool

Using the command line tool functions described above, we would make the following function calls to create the entries needed by our replication scenario:

Registering the collection

```
globus-replica-catalog –host <ldap URL for logical collection Collection1> \
  -manager <ldap DN of directory manager> -password <password file name> \
  -collection –create listCollectionNamesFile
```

Registering location A

```
globus-replica-catalog –host <ldap URL for logical collection Collection1> \
  -manager <ldap DN of directory manager> -password <password file name> \
  -location “locationA” –create “gridftp://Ahost.isi.edu:2222/nfs/path/on/A/” \
  listANamesFile
```

Registering location B

```
globus-replica-catalog –host <ldap URL for logical collection Collection1> \
  -manager <ldap DN of directory manager> -password <password file name> \
  -location “locationB” –create “ftp://Bhost.mcs.anl.gov:7777/nfs/path/on/B/”
\
  listBNamesFile
```

Registering logical file “file1”

```
globus-replica-catalog –host <ldap URL for logical collection Collection1> \
  -manager <ldap DN of directory manager> -password <password file name> \
  -logicalfile “file1” –create 104857600
```

Registering logical file “file2”

```
globus-replica-catalog –host <ldap URL for logical collection Collection1> \
  -manager <ldap DN of directory manager> -password <password file name> \
  -logicalfile “file2” –create 209715200
```

Registering logical file “file3”

```
globus-replica-catalog –host <ldap URL for logical collection Collection1> \
  -manager <ldap DN of directory manager> -password <password file name> \
  -logicalfile “file3” –create 314572800
```

Registering logical file “file4”

```
globus-replica-catalog –host <ldap URL for logical collection Collection1> \
  -manager <ldap DN of directory manager> -password <password file name> \
  -logicalfile “file4” –create 419430400
```

Registering logical file “file5”

```
globus-replica-catalog –host <ldap URL for logical collection Collection1> \
  -manager <ldap DN of directory manager> -password <password file name> \
  -logicalfile “file5” –create 524288000
```

Searching for the uc (Url Constructor) attribute of all locations that contain “file4” and “file5”

```
globus-replica-catalog –host <ldap URL for logical collection Collection1> \
  -manager <ldap DN of directory manager> -password <password file name> \
  -collection –find-locations namesToSearchFile –attributes uc
```

List the value of the size attribute of file “file2”

```
globus-replica-catalog –host <ldap URL for logical collection Collection1> \
  -manager <ldap DN of directory manager> -password <password file name> \
  -logicalfile “file2” –list-attributes size
```