Replication "API" Draft

replicateDataset

Purpose:

Query metadata service for a list of files to get for a replicated data node.

Discussion:

The function will be independent of any particular data movement agent (DMA). The first supported DMA is expected to be the Bulk Data Mover (BDM).

Input:

sourceGtwy: Gateway host running web services.

datasetId: Full identifier for dataset being requested; suitable for use with web service *getDatasetFiles()*.

dataMoverType: Indicator for format required by a supported DMA.

Output:

File on the local file system suitable as input to the data moving agent. This file will instruct the DMA to transfer all of the data files for one data set to the local disk.

Status: success/failure

updateReplicatedDataset

Purpose:

Query a gateway for names related to a data set to determine which files, if any, are needed to keep the replicated data synchronized with the source.

Discussion:

General approach: (1) Fetch thredds catalog from source data node and compare it to thredds catalog on local data node to identify which files to copy. (2) Prepare a file that the data movement agent (e.g. BDM) can use to initiate data transfers. Outstanding issues are discussed below.

Input:

sourceGtwy: Gateway host running web services.

datasetId: Full identifier for dataset being requested; suitable for use with web service *getDatasetFiles()*.

dataMoverType: Indicator for format required by a supported DMA.

Output:

File on the local file system suitable as input to the data moving agent. This file will only transfer files not present on the local file system.

Status: success/failure

publishReplicatedDataset

Purpose:

Push the metadata for a replicated data set to a gateway.

Discussion:

Provides thredds catalog from the source data node as input to the publishing client, which will use it for comparison/verification of the thredds catalog produced during publication. Other inputs to the function include a map file (or inventory) of the files, source dataset's gateway, the dataset ID, and an input flag for replication.

Input:

mapFile: Listing of files in dataset.
sourceGtwy: Gateway host running web services.
replicaFlag: Flag to pass to publication client to mark dataset as replicated.
datasetId: Full identifier for dataset being published.
sourceThredds: thredds catalog from the data node where the data set was copied.

Output:

Status: success/failure

Open Issues for replication client API:

Assumptions:

1) Restrictions on File Location: The output location on the local disk is assumed to be the current directory unless otherwise indicated.
2) Restrictions on Dataset ID: Although dataset IDs imply a hierarchy, the arguments defined here explicitly identify only one set of files. The replication client will not make use of the *getDatasetHierarchy()* web service.

For updateReplicatedDataset command:

1) What comparisons can/should we do between thredds catalogs? In upcoming version of publisher, the thredds catalog will have file ids and file version information. If files have the same name but source site has a newer version, then it needs to update the file at the mirror site.
2) Question of where we will get the URL to the thredds catalog on the data node: Currently don't have that information; would need to store the URL for the data node tomcat server/access point in the metadata catalog
3) If a source site is using checksums, then during publication at the source site, it will compute a checksum and compare to previous checksum; if different, treats file as a new file; version number will be associated with the file ID.
4) What if thredds catalog includes a hierarchical dataset with multiple levels? URL will be associated with a higher level in the hierarchy; wouldn't be a catalog associated with the dataset ID if it is in the middle of the hierarchy? If a gateway listing service returns a thredds catalog URL, it may return a null value for intermediate datasets in the hierarchy. Users get a dataset ID from browsing on the portal; client would have to figure out whether there is a thredds URL associated with the dataset or return an error.
5) Whether to include the mirror site thredds URL as an input argument. Maybe go to the replica gateway to get the thredds URL. Each data node may have a different thredds URL per data set. Query the local replica gateway to resolve(?).

For publishReplicatedDataset command:

1) May need to supply source thredds catalog to the publishing client. It is possible for information to be in the thredds catalog that is not necessarily contained in the map (inventory) file, e.g. some properties in the original thredds catalog may not be generated in the publication process itself; may need source thredds to make sure these are present at the mirror gateway.
2) At end of this function, the publisher creates a thredds catalog but does NOT want to push this immediately to thredds data server (TDS) or gateway. Wait until after the mirror site's thredds catalog has been compared to the source site's thredds catalog to ensure correctness of the replication operation. This can be done by the publisher separating these steps: produce a catalog, do comparison, then do publication if correctness is verified. This functionality already exists to some extent in the publisher, which creates thredds catalog first and then updates the TDS. May want to separate out update of TDS.