Infrastructure - Task #3077

Task # 3074 (Closed): Phase one implementaion for morpho to connect dataone services

Task # 3075 (Closed): Morpho needs new data file managment system and the local identifier authority.

Morpho file name convention and the mapping between the file name and identifier

2012-07-17 02:50 - Jing Tao

Status:	Closed	**Start date:**	2012-07-17
Priority:	Normal	**Due date:**	
Assignee:	Jing Tao	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	2013.10-Block.2.1		
Milestone:	None	**Story Points:**	
Product Version:	*		

Description

Currently, Morpho uses something like 100.1 as data/metadata file name.

Since the file name in the disk will be different to the package identifier, it only needs to be locally unique and we can still use the sequence number as local file name. But since we may allow users to copy external data file into the data file directory (we will make this directory transparent to users) in the future, using the sequence number as file name is problematic.

Since we can't use the identifier or some meaningful name as the file name, I propose to name a file base on the time stamp:

yyyy-mm-dd-hh-mm-ss-suffix.file-extention

suffix is a 2 digits random number.

file-extention:
metadata file -- .meta
data file -- .data
ORE file -- .ore
sysmeta file -- .sysmeta

In dataone api, every data/metadata file should have associated system metadata. Those two files should have same prefix, but have different extension.
Fox example, 2012-07-16-16-20-15-01.sysmeta contains the system metadata describing the data file 2012-07-16-16-20-15-01.data

History

#1 - 2012-07-17 02:55 - Jing Tao

For fileName-identifier mapping, we can use a java property text file to store the mapping. This file can locate at the root of the data directory and will not be treated as a data/metadata file.

#2 - 2012-07-17 16:02 - Matthew Jones

I think this approach might run into some troubles because of time conflicts. While its fine to have the date in the filename, we probably also need some sort of a key that is guaranteed unique and can be used as a surrogate for the PID. Maybe we could use a hash value or a UUID? And have you considered separating out the meta, ore, and data files into separate subdirectories? It would help overcome potential file system limits on number of files.

#3 - 2012-07-17 16:51 - Jing Tao

Separating out the meta, ore and data files into different subdirectories is a good idea. I am going to add this to Task #3076.

This is the file name, we only need its locally unique. Since we append a two digits random number to time stamp, it is hard to duplicate the file name. We may increase the digits to 4 for more safe. Of course, we can append a hash value or UUID to the time stamp. But the file name is too long.

If we want to transform this file name to the PID, we can combine a unique id of the computer with the file name. The unique id of the computer can

be the mac address, the motherboard id or a UUID.

The problem to use the mac address is that users can exchange network cards between computers. The problem to use motherboard id is that java doesn't have elegant way to get it.

#4 - 2012-07-18 21:02 - Ben Leinfelder

I'm not so keen on using the dates as file name parts and would much rather see hashcodes or checksums of the PIDs that we can (hopefully) rely on to be collision free (SHA1-1 rather than MD5). The implementation for String.hashCode() has been defined and codified, but in general hasCode() methods should not be relied upon to produce the same value except during subsequent invocations in the same JVM.

#5 - 2012-07-29 23:19 - Jing Tao

Some thought about the identifier-filename mapping file:
It will be stored at /user-home/.morpho/profiles/profile-name.
It is a csv file. The first element is the identifier, the others are file locations. One identifier may have more one location - local store and cache directory.
The object to represent the mapping will be HashMap>

Issue: is comma a safe separator?

#6 - 2012-08-02 16:08 - Jing Tao

Since we may need more information than just an identifier mapping a file name, we may need a xml property file to hold the information.
The file may look like:

doi:10.6085/AA/CMRX00_XXXITBDXLSR02_20060618.50.5
/my-data/morpho/metadata/12345
/home/john/.morpho/profiles/john/cache/metadata/12345
a532fcb3d1ca297fdd3dc1cc5fa6b1a7

#7 - 2012-10-03 14:41 - Ben Leinfelder

- Description updated

- Assignee set to Jing Tao

#8 - 2012-10-03 15:06 - Ben Leinfelder

I think this can be much simpler and made as a reusable utility class. To me, the only important aspects are: "identifier=filepath" and the xml syntax is very verbose for that. In the morpho-class-diagram, take a look at FileSystemDataStore -- I think we can use instances of this class to support the current (and future) uses in Morpho.

For our current Morpho needs we'd have these instances available for looking up/setting the files we work with:

FileSystemDataStore data; // EML and data
FileSystemDataStore cache; // remote EML and data not yet explicitly saved
FileSystemDataStore incomplete; //wizard stopped mid-way through EML generation
FileSystemDataStore temporary; // created locally, but not yet explicitly saved

DataONE would als require:

FileSystemDataStore systemMetadata; // stores system metadata as XML serialization
FileSystemDataStore ore; //I'd argue that these should just be part of the existing store for EML/data files

For the initial implementation, I would use a simple Properties file (assuming the identifier "key" can safely include characters like ":" and "=" without

throwing the Properties parser off. I think a DB implementation of the Map adds a lot of overhead.

We'd need the above structure for each user profile since (currently in Morpho) it's entirely possible for different profiles to work on packages with the same Identifier and not overwrite the content.

#9 - 2012-10-03 15:57 - Jing Tao

Some comments:
1. "To me, the only important aspects are: "identifier=filepath"". However, identifier can contain illegal characters for a file path. This is the reason we need a map between identifier and file path.
2. "For the initial implementation, I would use a simple Properties file". If we use the property file, we assume the id:file path = 1:1. However the relationship is 1:* (multiple, not including 0) since a id can be stored in more than one data stores. For example, a identifier can be stored in both local and cache data stores. This the reason that i discarded property file and use a xml file.

#10 - 2012-10-11 15:33 - Dave Vieglais

- Target version changed from Sprint-2012.37-Block.5.3 to Sprint-2012.41-Block.6.1

#11 - 2012-10-14 14:45 - Ben Leinfelder

- translation missing: en.field_remaining_hours set to 0.0

- Status changed from New to Closed

This has been implemented. For file generation in each directory, we are using the File.createTempFile() mechanism that guarantees no collision in the given directory. We continue to have different directories for each kind of storage. Right now that is limited to:
data (EML and data files saved locally)
cache (EML and datafiles opened from Metacat)
temp (serializations of files before upload, I believe)
queries (any saved pathquery snippets)
incomplete (EML files that were generated with the wizard without finishing all wizard screens)

#12 - 2012-10-24 18:20 - Ben Leinfelder

- Target version changed from Sprint-2012.41-Block.6.1 to Sprint-2012.44-Block.6.2

#13 - 2012-12-12 16:51 - Chris Jones

- Target version changed from Sprint-2012.44-Block.6.2 to Sprint-2012.50-Block.6.4

#14 - 2013-03-01 18:33 - Ben Leinfelder

- Target version changed from Sprint-2012.50-Block.6.4 to 2013.10-Block.2.1