Infrastructure - Story #3448

add (client) methods to traverse obsoletes chain

2013-01-03 22:59 - Rob Nahf

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Rob Nahf	**% Done:**	100%
Category:	d1_libclient_java	**Estimated time:**	0.00 hour
Target version:			
Story Points:			

Description

clients working with other people's data objects could use a convenient way to know if they have the latest version of a file - for example, corrected data instead of the original.

At least at the client side, it should be possible to use the obsoletes and obsoletedBy tags in the systemMetadata to get the latest version, and to get the entire obsoletes chain, ideally including the publication dates of each.

Consider an ObsoletesChain class, or a simple sortable map (TreeMap ?) to hold the chain. Advantages of an ObsoletesChain class would be the ability to add comparator and other "versionAsOf(Date date)" methods

Subtasks:

Task # 3540: design and build getObsoletesChain method	**Closed**
Task # 3541: design and build the ObsoletesChain object	**Closed**

Related issues:

Related to DataONE R Client - Feature #3639: Create DataPackage from ID of me...	**Closed**	**2013-03-05**	**2014-09-10**

History

#1 - 2013-01-31 22:48 - Rob Nahf

- % Done changed from 0 to 80

- Status changed from New to In Progress

designed and built ObsoletesChain data object to hold the obsoletes chain path traversal.
designed and built the D1Client.getObsoletesChain method

#2 - 2013-03-01 18:39 - Ben Leinfelder

I'd argue for adding a server-side method for retrieving this information. We've found that long revision chains take a long time to traverse in Morpho because we have to recursively fetch the SystemMetadata for each revision.

#3 - 2013-03-01 19:01 - Rob Nahf

- Target version changed from 2013.2-Block.1.1 to 2013.10-Block.2.1

- Due date changed from 2013-01-31 to 2013-03-16

#4 - 2013-03-07 00:28 - Rob Nahf

That's a good idea. The client-side implementation can work as a stop-gap solution, until the infrastructure is built-out. There is already demand in the R-Client for getting to latest packages. (#3639).

Another optimization for the client-side implementation is to lazy-load the obsoletes chain based on the method called from it - if someone wants only the latest version of something, avoid fetching predecessors until needed, etc.

#5 - 2013-05-23 18:00 - Rob Nahf

- Target version changed from 2013.10-Block.2.1 to 2013.20-Block.3.2

- Due date changed from 2013-03-16 to 2013-05-25

#6 - 2013-05-31 16:24 - Rob Nahf

- Due date changed from 2013-05-25 to 2013-06-08

- Target version changed from 2013.20-Block.3.2 to 2013.22-Block.3.3

#7 - 2013-08-07 19:12 - Rob Nahf

- Target version changed from 2013.22-Block.3.3 to 2013.30-Block.4.3

- Due date changed from 2013-06-08 to 2013-08-10

#8 - 2013-08-08 20:39 - Rob Nahf

- File ObsoletesChainTest.java added

- File ObsoletesChain.java added

- File D1Client.java added

Need to finish this implementation in trunk, but currently it's not ready - some of the ObsoletesChain methods (like getLatestVersion()) need to be made aware of NotAuthorized situations that block getting to the end of a chain. parking the code in the attached files.

#9 - 2013-08-14 16:21 - Rob Nahf

client side implementation falls short because of potential access-policy differences, so users aren't always able to navigate the entire chain. I suggest leaving this out of libclient_java, in favor of a server-side implementation.

#10 - 2013-08-14 18:37 - Rob Nahf

- Start date deleted (2013-01-31)

- Target version deleted (2013.30-Block.4.3)

- Due date deleted (2013-08-10)

move to backlogs until server side vs. client side approach can be decided.

#11 - 2014-10-01 23:05 - Rob Nahf

- Status changed from In Progress to Closed

Files

ObsoletesChain.java	4.47 KB	2013-08-08	Rob Nahf
ObsoletesChainTest.java	4.19 KB	2013-08-08	Rob Nahf
D1Client.java	13.3 KB	2013-08-08	Rob Nahf