Metacat - Bug #7178

MNodeService.getPackage() takes too long for large packages

03/29/2017 07:22 AM - Chris Jones

Status:	New	**Start date:**	03/29/2017
Priority:	Normal	**Due date:**	
Assignee:	Chris Jones	**% Done:**	0%
Category:	metacat	**Estimated time:**	0.00 hour
Target version:		**Spent time:**	0.00 hour
Bugzilla-Id:			

Description

When users click on the Download All button in MetacatUI, we call MN.getPackage() to zip up the members, create HTML and PDF metadata, etc. For very large packages, the packaging time is far too long for a decent user experience. To address this, we may need to provide some sort of progress API call that allows the client to get estimated packaging time and provide a progress bar for the user. Also, getPackage() uses File.createTempFile() to copy contents into a single directory tree for zipping (really BagIt bagging). This doesn't scale well for large packages (GBs). We can explore a few strategies to mitigate this. One that comes to mind is using hard symbolic links to the original data files in the directory tree rather than copying them. This needs some thought, but ultimately we need to speed up the packaging process for large packages.

History

#1 - 03/29/2017 09:18 AM - ben leinfelder

If you can figure out how to reference files rather than copy them, I FULLY support that! I suppose the MN.getPackage() method just needs to pull directly from the Metacat filesystem and instead of File.createTempFile() we'd use Files.createSymbolicLink() or Files.createLink() as described here:
https://docs.oracle.com/javase/tutorial/essential/io/links.html

The one drawback is that we wouldn't get to benefit from the access rule checking that is inherent in the current implementation that just repeatedly calls MN.get() to fetch the bytes of the data.