Metacat - Bug #3464

Replicate access rules in replication

07/28/2008 10:36 AM - Jing Tao

Status:	Resolved		**Start date:**	07/28/2008
Priority:	Immediate		**Due date:**	
Assignee:	ben leinfelder		**% Done:**	0%
Category:	metacat		**Estimated time:**	0.00 hour
Target version:	1.9		**Spent time:**	0.00 hour
Bugzilla-Id:	3464			

Description

Currently, metacat doesn't replicate access rules in xml_access table.
For eml documents, it wouldn't be a problem since eml itself has the access rules. However, for none-eml documents, it can be problem:
A host was inserted a FGDC document and user uses setAccess API to make this public readable. So in host A, the document is public readable. However, in host B, which got the replicated copy, the document is not public readable since the document itself doesn't have the access rules and both timed and force replication doesn't replicate access rules.

Here is the plan:
1. setAccess method will be added into the listener of force replication action.
2. Timed replication will replicate xml_access table as well.

History

#1 - 07/28/2008 11:55 AM - Matt Jones

This is a problem for EML documents as well, especially if someone changes the access rules using setAccess and then replicaiton doesn't respect that. Also, it seems that all of the different replication methods should be replicating these rules.

#2 - 10/07/2008 04:42 PM - ben leinfelder

committed replication changes:
-document access rules will be replicated
-data access rules will be replicated
-setting access via the metacat servlet (action=setaccess) will result in a forced replication for that docid (data or xml)

now i need to test it!

#3 - 10/13/2008 01:09 PM - ben leinfelder

set up replication between my two local metacats. tested:
-forced replication of document due to access control change
-forced replication because of document update

also: duplicate access control rules will not be persisted in the xml_access table

#4 - 11/24/2008 08:22 AM - ben leinfelder

Mike has run through testing scenarios for ACL replication with no hiccups.
Closing this bug.

#5 - 03/27/2013 02:23 PM - Redmine Admin

Original Bugzilla ID was 3464