

Infrastructure - Bug #7371

A systemMetadata update that changes the formatID from DATA to METADATA or RESOURCE_MAP will not upload the object to the CN

2015-09-21 21:54 - Rob Nahf

Status:	New	Start date:	2015-09-21
Priority:	Normal	Due date:	
Assignee:	Robert Waltz	% Done:	0%
Category:	d1_cn_service	Estimated time:	0.00 hour
Target version:	Release Backlog	Story Points:	
Milestone:	None		
Product Version:	*		
Description			
With V2, systemMetadata.formatID is allowed to change. If the formatID changes from a DATA format to another one after the initial harvest, d1_sync doesn't have a cn method that will persist the object if the systemmetadata is already registered. cn.create could be modified, but it might be better to define a new v2 API method (cn.refreshObject?)			
Related issues:			
Related to Infrastructure - Story #7364: update the sync logic for basic v1-v...		Closed	2015-09-22
Related to Infrastructure - Task #3787: CN should detect changed sysmeta and ...		Closed	2013-06-04

Associated revisions

Revision 16416 - 2015-09-23 06:33 - Rob Nahf

refs: #7364, #7371. serialVersion handling changed to ignore MN values, but use the one already in the CN version of it, or initialize to one. Now logging changes to formatID that result in a FormatType change.

Revision 16416 - 2015-09-23 06:33 - Rob Nahf

refs: #7364, #7371. serialVersion handling changed to ignore MN values, but use the one already in the CN version of it, or initialize to one. Now logging changes to formatID that result in a FormatType change.

History

#1 - 2015-09-21 21:55 - Rob Nahf

- Related to Story #7364: update the sync logic for basic v1-v2 interoperability added

#2 - 2015-09-21 21:55 - Rob Nahf

- Related to Task #3787: CN should detect changed sysmeta and resync objects when needed added

#3 - 2015-09-22 19:45 - Robert Waltz

If systemMetadata is updated, CNCORE.updateSystemMetadata, and Metacat notes that the formatID has changes, which in turn changes the content residing in or on Metacat, then Metacat should handle the change. Not d1_cn_service or cn synchronization.

Other systems, cn-rest-service or cn synchronization should not be responsible for deleting, creating or otherwise modifying the underlying storage mechanisms of Metacat.

any added method, such as 'cn.refreshObject', would simply perform what metacat should do when CNCORE.updateSystemMetadata is called.

#4 - 2015-09-22 19:48 - Matthew Jones

This sounds like a private internal function of the CN service, not a public API method.

#5 - 2015-09-22 20:38 - Rob Nahf

If we didn't already use restricted CN API methods to upload and persist content to the CN, I would tend to agree. However, responsibility for whether or not to upload and persist content on the CN exists wholly within d1_sync (and accomplished via API calls), so I think we do better to avoid spreading responsibility to other packages and end up with more than one component responsible for the same thing.

Currently, we use restricted CN API methods to persist new systemMetadata (cn.registerSystemMetadata), update systemMetadata (cn.updateSystemMetadata), and create new content (cn.create) on the CN. For v1 content, the cn.setAccessPolicy, cn.obsoletedBy, cn.archive methods are used. Within sync, there is robust logic for retrying content uploads.

#6 - 2015-09-22 21:47 - Rob Nahf

- Target version changed from CCI-2.0.0 to Release Backlog

discussed in extra maintenance meeting today. Decided that since we have a "manual intervention" process, this is not a priority for V2.1. Moving to the release backlog.

We did not discuss thoroughly the issue of whether it should be an API method or otherwise.

Synchronization will create a log entry, however, when formatID changes and would require uploading the object to the CN.