

Infrastructure - Story #3667

MN.update() should set archive field for the obsoleted object

2013-03-19 18:50 - Rob Nahf

<b>Status:</b>	Rejected	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Dave Vieglais	<b>% Done:</b>	100%
<b>Category:</b>	Documentation	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>			
<b>Story Points:</b>			
<b>Description</b>			
obsoleted data should not clutter search results, but the current MN.update() method specification only mentions setting the appropriate obsoletes / obsoletedBy fields.			
MN.update should also ensure that the obsoleted objects are also archived.			
documentation and implementations need to reflect these changes, as well as existing content in production. No change to the indexer seems to be needed.			
the IRC discussion:			
[12:09pm] rob: Chris, did you want me to create a ticket for MN's archiving obsoleted objects during MN.update?			
[12:12pm] chris: sure - just a ticket for dave to update the docs to say that update() 'SHOULD' call archive()			
[12:12pm] chris: thx			
[12:12pm] rob: sure.			
[12:13pm] rob: do we want to also have the CN's set the archived flag for all the obsoleted data in production now?			
[12:14pm] rob: or have member nodes do the work themselves?			
[12:14pm] chris: hmm. good question.			
[12:14pm] chris: one sec			
[12:14pm] rob: (good test of the MN.systemMeatadataChanged() method)			
[12:16pm] rob: we'll also have to notify Tier1 (and 2) Nodes that they may need to change their implementation of update, too. I'll need to write an integration for this ...			
[12:17pm] chris: interesting - i'm looking at Metacat's update(), and am not seeing a call to archive()			
[12:18pm] chris: so, I think this needs to be done on the MN side			
[12:19pm] chris: we'd fix update(), then on the 2.0.7 release of Metacat, each site would go through an upgrade process to find obsoleted pids that are yet to be archived, then archive them			
[12:19pm] chris: then the CN would pick up on all that			
[12:20pm] chris: so, would you make it a story, and assign a subtask to Ben or me to update Metacat? And perhaps Roger for GMN if it doesn't archive()?			
[12:20pm] rob: aye-aye cap'n			
[12:20pm] skye: if we really want obsolete to be kept out of the index via update/archive, shouldn't the CN be enforcing the archive flag is set. otherwise its a MN policy decision?			
[12:21pm] chris: hmm. good point. seems like we need both			
[12:22pm] robert: another point in the conversation about control of sysMeta			
[12:22pm] skye: true			
[12:23pm] chris: this could be handled in d1_sync or in Metacat, and it could throw InvalidSystemMetadata if obsoletedBy is present but archived is not			
[12:23pm] chris: are we sure there are no edge cases on this though?			
[12:25pm] robert: why we'd want obsoletedBy set but not archived?			
[12:25pm] chris: i can't think of any scenario			
[12:27pm] rob: me neither.			
[12:28pm] chris: robert - in sync, when you are updating an object, do you call create()?			
[12:28pm] chris: CN.create() that is			
[12:29pm] chris: i ask, because i see this in CNodeService.create():			
[12:29pm] chris: sysmeta.setArchived(false); // this is a create op, not update			
[12:43pm]			
and earlier...			
[11:53am] skye: yeah...so anyways doesn't sound like any change to indexing to me			
[11:55am] rob: cool, then you don't know of any use case for keeping these-things-that-will-now-get-archived-automatically in the			

search index, do you?

[11:56am] rob: (aka: things-that-were-once-only-obsolete)

[11:57am] skye: well I think you mentioned the motivation to have the obsolete in the search - it makes finding the package information for an obsolete document hard

[11:57am] skye: if they are not in the index

[11:58am] skye: but if they can trace the obsolete chain to the current version of the doc/pid, then they can discover the package info of the current version of the objects

[11:58am] rob: yes, exactly - that's an important use case.

[11:59am] rob: and they would then have to travel back in time along the package obsoletes chain to get the relationships for obsolete items.

[11:59am] rob: not impossible, but not fun either

[12:00pm] skye: right

[12:00pm] rob: I think it's a clearer implementation, though

[12:01pm] rob: people doing search don't have to be so "in-the-know"

[12:01pm] skye: yeah after thinking about it, having just one attribute control index/search visibility is good

[12:01pm] skye: well they do

[12:01pm] skye: still

[12:02pm] skye: know they have to know that obsolete documents are not in the index - not sure that will be obvious to all people

[12:02pm] skye: now

[12:03pm] skye: i can imagine support questions - i can find my object using /resolve and /object but not with search

[12:03pm] rob: yes - a good question for ask.dataone.org

[12:05pm] skye: this change would require our production MN to archive all their current obsolete docs?

[12:06pm] skye: or something to be pushed out from the CN to the MN?

[12:07pm] rob: good question

Subtasks:	
Task # 3668: change MN.update to reflect archive field of obsolete to be set by update	Rejected
Task # 3669: metacat MN.update() should set archived for obsolete	Rejected
Task # 3670: GMN MN.update() should set archived for obsolete	Rejected
Task # 3671: sync return InvalidSysMeta for obsolete but not archived	Rejected
Task # 3672: integration test for checking archived upon update	Rejected
Task # 3673: implement MN.update() method change in existing member nodes.	Rejected
Task # 3674: archive obsolete objects in production	Rejected

## History

### #1 - 2013-03-19 19:02 - Rob Nahf

- Description updated

### #2 - 2013-03-22 17:47 - Rob Nahf

- Status changed from New to Rejected

this story was to solve the problem that archived content was not in cn/solr index, but obsolete objects were - leading to inconsistent access to package relationships for non-current objects.

The solution asserted in this story was to add the archived field to update - so that all obsolete data was also archived, but this took away the ability to access package relationships of all objects - a perceived need.

Instead the semantics for 'obsoleteBy' will imply archived, and the cn/solr index appropriately adjusted. The current situation will remain (archived not indexed, obsolete objects indexed) until another service is in place to handle relationship navigation.