

Metacat - Bug #7083

Metadata/data objects which have obsoletedBy field ignore the resource map index

08/08/2016 02:17 PM - Jing Tao

Status:	Closed	Start date:	08/08/2016
Priority:	Normal	Due date:	
Assignee:	Jing Tao	% Done:	0%
Category:	index	Estimated time:	0.00 hour
Target version:	2.7.2	Spent time:	0.00 hour
Bugzilla-Id:			

Description

Hi Bryce:

I looked at the index of the 16 objects and found 5 of them don't have the value of resource\_map\_urn:uuid:2e3c8c4c-e606-4710-b321-8edc4d506b0a at the resourceMap element:

urn%3Auuid%3A0f64673d-d270-411f-a5ed-98351d3d9450  
urn%3Auuid%3A12c0ab6a-5eb3-43de-a16c-e71acaeb9817  
urn%3Auuid%3A45ee065f-746e-4780-872b-d98cabe0ad7  
urn%3Auuid%3Aae90efa8-3cf5-4ff9-9637-c7be28b06541  
urn%3Auuid%3Acedeb0b-6bdb-4853-ba2a-6e88321ea4d5

So this is the reason you only get 11 documents when you query this resource map value.

And all of the five objects have the field "obsoletedBy" and the other 11 object don't have the field.

The reason why I looked at the field "obsoletedBy" is I recently found that there was a bug in the d1\_cn\_index\_processor component - when you index a resource map, the component in the resource map will ignore the resource map if it has the "obsoletedBy" field. So this issue sounds like the reflection of this bug.

I will look at the metacat index code to make sure.

Thanks,

Jing

On 8/8/16 12:13 PM, Bryce Mecum wrote:

So @scng got a hold of me to ask about strange behavior where there package table on two dataset pages are not showing the right number of files. This is a write up of what she told me and what I found so that someone else, [Jessica Couture](#) or [Chris Jones](#) can see about addressing it. This is a blocker on Bill Simpson's ticket RT12930.

This applies to two packages:

O-Buoy 8 (needs link)  
O-Buoy 15

These two packages were recently updated to make them editable (adding otherEntity elements to the EML) by @scng using the R package.

If you look at O-Buoy 15, you'll see ten data objects in the package. However, the R @scng wrote intended to add 15 data objects to the package. If you look at the resource map, resource\_map\_urn:uuid:2e3c8c4c-e606-4710-b321-8edc4d506b0a, you'll see it aggregates+documents 16 PIDs (metadata + 15 data):

Here's an invalid and abridged section from the resource map, converted to Turtle format before pasting here:

...

ore:aggregates <<https://cn.dataone.org/cn/v2/resolve/urn%3Auuid%3A0f64673d-d270-411f-a5ed-98351d3d9450>>  
<<https://cn.dataone.org/cn/v2/resolve/urn%3Auuid%3A12c0ab6a-5eb3-43de-a16c-e71acaeb9817>>  
<<https://cn.dataone.org/cn/v2/resolve/urn%3Auuid%3A1584c53e-3d5c-4b70-9bf6-1033de8e2fd1>>  
<<https://cn.dataone.org/cn/v2/resolve/urn%3Auuid%3A1c2d1c50-4d79-4fe5-b650-024e63818336>>  
<<https://cn.dataone.org/cn/v2/resolve/urn%3Auuid%3A2e3c8c4c-e606-4710-b321-8edc4d506b0a>>  
<<https://cn.dataone.org/cn/v2/resolve/urn%3Auuid%3A30a3a76c-c965-4594-8cfd-c652d46ebbe5>>

<<https://cn.dataone.org/cn/v2/resolve/urn%3Auuid%3A40d6e8e4-83eb-4579-8b00-90bf28282769>>  
<<https://cn.dataone.org/cn/v2/resolve/urn%3Auuid%3A45ee065f-746e-4780-872b-d98cabe0ad7>>  
<<https://cn.dataone.org/cn/v2/resolve/urn%3Auuid%3A4eb92d77-19f4-4a3a-8468-4022926ea4e2>>  
<<https://cn.dataone.org/cn/v2/resolve/urn%3Auuid%3A6d57e765-32a0-4a3e-ba12-5e681f92b7e5>>  
<<https://cn.dataone.org/cn/v2/resolve/urn%3Auuid%3A73926857-7d7c-4a6e-bce3-1556bd98df01>>  
<<https://cn.dataone.org/cn/v2/resolve/urn%3Auuid%3A770eb22d-88bb-4c6f-9016-283f4ff7a518>>  
<<https://cn.dataone.org/cn/v2/resolve/urn%3Auuid%3A8539eac4-21f5-4a3a-8c0a-5ad7249cf38c>>  
<<https://cn.dataone.org/cn/v2/resolve/urn%3Auuid%3Aae90efa8-3cf5-4ff9-9637-c7be28b06541>>  
<<https://cn.dataone.org/cn/v2/resolve/urn%3Auuid%3Aaccebed0b-6bdb-4853-ba2a-6e88321ea4d5>>  
<<https://cn.dataone.org/cn/v2/resolve/urn%3Auuid%3Ad54c9d42-99ce-415b-ac7c-a2b3498eb7af>> ;

...

So it looks like the Resource Map is correct which makes sense because it was generated using the R package.

The package view uses the Solr query resourceMap:{RESOURCE\_MAP} to fill in the table. If you run this query you see the 11 objects, not 16. This explains the table view not showing all the files.

If you look at the documents section of the metadata object's Solr doc, you'll see the 16 objects it documents (itself + 15 data objects).

So what's going on here? Am I wrong to think that it's just the index that is showing the wrong information?

I have forced a reindex with no change  
I have not checked the arctica logs for any errors

#### Related issues:

Has duplicate Metacat - Bug #7093: Metacat-index is not indexing all package ...

New

08/26/2016

#### History

##### #1 - 08/18/2016 02:40 PM - Jing Tao

- Status changed from New to Closed

It was fixed in the d1\_cn\_index\_processor component. I created a new 2.2.1 tag and built it on Jenkons. Please see:

<https://redmine.dataone.org/issues/7867>

In the metacat-index pom.xml file, the dependency of d1\_cn\_index\_processor was modified to the version 2.2.1.

##### #2 - 08/26/2016 11:01 AM - Chris Jones

- Has duplicate Bug #7093: Metacat-index is not indexing all package members correctly added