

## Infrastructure - Bug #7284

### MN.updateSystemMetadata() should not allow pid chain inconsistency

2015-07-30 18:50 - Andrei Buium

<b>Status:</b>	Closed	<b>Start date:</b>	2015-07-30
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Jing Tao	<b>% Done:</b>	100%
<b>Category:</b>	Metacat	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	CCI-2.2.0	<b>Story Points:</b>	
<b>Milestone:</b>	None		
<b>Product Version:</b>	*		

#### Description

Scenario:

a pid chain:

P1 <---> P2

P1 obsoletedBy P2

P2 obsoletes P1

MN.updateSystemMetadata() is called and sets

P1.obsoletedBy = P3

This would cause an inconsistency / branch (since P2 still obsoletes P1).

This is currently allowed, but should throw an InvalidSystemMetadata exception.

#### History

##### #1 - 2015-07-30 21:19 - Jing Tao

I just read the documentation of the system metadata and found both "obsoletedBy" and "obsoletes" are mutable. So we should allow the change happen. But we can't allow to create a branch like the scenario which Andrei described. So we have two options:

1. Totally trust user - we assume the user will call another mn.updateSystemMetadata(p2); it will set p2 obsoletes null. Then the new branch will disappear.

1. When mn.updateSystemMetadata change the field of obsoletedBy, it also will change the system metadata of pid in the obsoletedBy field (this case is p2). The "obsoletes" of p2 will be set to null. If p2 doesn't exist in the node, the update still succeed. If the update on p2 system metadata fails, the method call fails as well. Same scenario apply to "obsoletes" too.

##### #2 - 2015-07-30 21:25 - Andrei Buium

Option (1) with completely trusting the user makes me nervous. I was thinking we could block the change and pass back a message along the lines of "To set the obsoletedBy value, you need to first clear the obsoletes value of this other pid, to avoid inconsistencies".

Option (2) sounds better to me. It's more complicated logic, but also more user-proof.

##### #3 - 2015-08-04 23:32 - Jing Tao

The "obsoletedBy" and "obsoletes" only can be set once. So the above resetting will fail.

##### #4 - 2015-08-04 23:40 - Jing Tao

- Status changed from New to Closed

- % Done changed from 0 to 100

Andrei tested it and it worked as we think.

**#5 - 2016-02-18 22:49 - Peter Slaughter**

- Status changed from Closed to In Progress
- % Done changed from 100 to 30
- Target version changed from CCI-2.0.0 to CCI-2.1.0

I'm reopening this issue as it appears that MN.updateSystemMetadata doesn't ensure consistency between the obsoleting and obsoleted sysmetas. On mn-stage-ucsb-2 pid2 sysmeta was updated to obsolete pid1, which did update pid2 system. However pid1 sysmeta should have been updated so that obsoletedBy -> pid2, but pid1 sysmeta is not updated.

Here is pid1 sysmeta: <https://mn-stage-ucsb-2.test.dataone.org/knb/d1/mn/v2/meta/urn:uuid:d25dfb45-610a-4303-b320-e727aa1f15d4>

and pid2 sysmeta: <https://mn-stage-ucsb-2.test.dataone.org/knb/d1/mn/v2/meta/urn:uuid:94c30aec-6781-4ac1-94f1-9682589b8341>

**#6 - 2016-02-19 18:07 - Jing Tao**

Hi Peter:

Your request is different to this ticket. The purpose of this ticket is prevent from generating a branch:

- p1: p1 obsoletedBy p2
- p2: p2 obsoletes p1.

But when you update system metadata of p3, you can't set p3 obsoletes p1 since this will generate branch.

In our original design, we don't have the feature you requested (automatically update the system metadata of p1 by setting obsoletedBy), you have to call the updateSystemMetadata on p1 as well in order to set the obsoletedBy field.

**#7 - 2016-03-25 02:33 - Dave Vieglais**

- Target version changed from CCI-2.1.0 to CCI-2.2.0

**#8 - 2016-03-25 04:26 - Jing Tao**

- % Done changed from 30 to 100
- Status changed from In Progress to Closed