Metacat - Bug #2469

DocumentImpl.buildIndex() does not index XPaths with attributes correctly

06/22/2006 03:15 PM - Chris Jones

Status:	Resolved	**Start date:**	06/22/2006
Priority:	Immediate	**Due date:**	
Assignee:	Saurabh Garg	**% Done:**	0%
Category:	metacat	**Estimated time:**	0.00 hour
Target version:	1.7	**Spent time:**	0.00 hour
Bugzilla-Id:	2469		

Description

A 1.6.x metacat installation that indexes paths from the xml_nodes table into the xml_path_index table sets the xml_path_index.path column correctly, but sets the xml_path_index.nodedata incorrectly for ATTRIBUTE nodes. This results in searches that return an incorrect subset of documents because the xml_path_index table doesn't reflect the true values in xml_nodes.

For example, an EML 2.0.1 document with a packageId attribute:

```
<eml:eml xmlns:eml="eml://ecoinformatics.org/eml-2.0.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
packageId="ALEXXX_015MTBD003R00_19990906.50.1"
scope="system" system="knb"
xsi:schemaLocation="eml://ecoinformatics.org/eml-2.0.1 eml.xsd">
<dataset scope="document">
<shortName>PISCO moored temperature, ALE</shortName>
```

```
... etc ...
  &lt;/dataset&gt;
&lt;/eml:eml&gt;
```

will contain an indexed record in xml_path_index with the following columns:

docid: ALEXXX_015MTBD003R00_19990906.50
path: /eml/@packageId
nodedata: PISCO moored temperature, ALE

rather than:

docid: ALEXXX_015MTBD003R00_19990906.50
path: /eml/@packageId
nodedata: ALEXXX_015MTBD003R00_19990906.50.1

It seems that the nodedata for the attribute is set to the node value of the next leaf node, in this case the /eml:eml/dataset/shortName field.

This also occurs for other attributes that are indexed in the document, such as /eml:eml/dataset/coverage/geographicCoverage/@id (which has a value of 'ALE')

The above @id will have an indexed value set to the geographicDescription value found in /eml:eml/dataset/coverage/geographicCoverage/geographicDescription (not 'ALE' as above)

Related issues:

Blocked by Metacat - Bug #2496: Wrong values indexed when empty tags are indexed	**Closed**	**07/20/2006**

History

#1 - 01/18/2007 03:31 PM - Chris Jones

This issue was being caused by offset indexing that involved 3 methods in DocumentImpl.java:

buildIndex()
traverseParents()
updatePathIndex()

The problem arose when the node data for ATTRIBUTE nodes and ELEMENT nodes is stored in different locations within the xml_nodes table. ATTRIBUTES store
their own data in a column of the same record, whereas ELEMENTS store their node data in a child TEXT node (an entirely different node record altogether, on linked by the parent node id relationship).

Now, these methods work together to create a HashMap of PathIndexEntry objects that associate the correct node data, path, and node id to be indexed, regardless of whether it is an ATTRIBUTE or an ELEMENT that is being indexed. Testing needs to be done.

Note: all metacat databases installed prior to this fix should run the action=buildindex task in order to remove all of the erroneous entries from before and create new path and node indices.

#2 - 01/25/2007 10:13 AM - Matt Jones

Fixed the implementation of the buildIndex function which was not working for new document insertions. A previous fix in updatePathIndex for ATTRIBUTE data inadvertantly caused a foreign key duplication exception for insertions of ELEMENT nodes when multiple relative paths exist. This fix simply reverts to the old behaviour of allowing the primary key of xml_path_index to be set using its sequence instead of manually matching it to the xml_nodes.nodeid (which the current code did and which caused the duplicate key problem). See bug 2469 for related details regaring the indexing changes.

#3 - 02/15/2007 10:55 AM - Chris Jones

As a continued fix for http://bugzilla.ecoinformatics.org/show_bug.cgi?id=2469, I've fixed the indexing implementation in both buildIndex() and traverseParents(). Duane pointed out that the incorrect parent node ids were being indexed in xml_path_index, causing some stylesheets to render metadata incorrectly. I've changed buildindex() to index the correct parent id, and have changed traverseParents() to only index paths that actually contain node data (some TEXT nodes exist with empty data).

#4 - 02/15/2007 02:59 PM - Chris Jones

One more patch for bug #2469:
Although the correct parentid values were being indexed in xml_path_index for leaf node xpaths, they were still incorrect for relative and absolute paths. This patch modifies traverseParents() and changes the parent node id to be indexed to that of the leaf node, no matter if the path is a leaf, relative, or absolute. Thanks for catching this Duane.

#5 - 03/27/2013 02:20 PM - Redmine Admin

Original Bugzilla ID was 2469