Metacat - Bug #1427

xml_index constrains depth of paths that can be inserted

03/30/2004 02:20 PM - Matt Jones

Status:	Resolved	**Start date:**	03/30/2004
Priority:	Immediate	**Due date:**	
Assignee:	Matt Jones	**% Done:**	0%
Category:	metacat	**Estimated time:**	0.00 hour
Target version:	1.4	**Spent time:**	0.00 hour
Bugzilla-Id:	1427		

Description

When an XML document contains a deeply nested structure, metacat accepts the document for storage in xml_nodes, but during the subsequent indexing phase, it throws an exception because the composite paths to the deep nodes are too long to fit in the space allocated for the paths in the column in the xml_index table. This column was limited to a a few hundred characters so that it is indexable (Oracle had a limit on the total indexable width of columns).

These problems were discovered and reported by Wade Sheldon (GCE LTER) when he submitted EML documents with fully filled out taxonomic coverage entries. We definitely need to support realistically filled out EML documents.

So, two possible solutions:
1) make the column much wider
-- this is a partial solution, because the column still might not be big
enough for very deep docs or docs with long element names
-- if its wider, it may not be indexable, which is why it exists
2) eliminate the dependency on the xml_index table altogether
-- the recursive search needed isn't that much slower, and may not be
slower at all as we tune the database
-- insert/update/delete should be MUCH faster
-- simpler database structure

We have decided to pursue (2) above because of the advantages listed. Rather than completely removing the xml_index code, we are going to make it an option whether or not it is used, but by default ship with it turned off.

History

#1 - 03/30/2004 02:26 PM - Matt Jones

The solution(2) above has been implemented and checked into CVS. Now, the metacat.properties files has a new option called "usexmlindex" that defaults to "false". If false, then all queries are done using recursive queries against the xml_nodes table, and insert, update, and delete actions all ignore the xml_index table. If usexmlindex is set to "true", the old behavior of maintaining paths in xml_index is preserved and the queries use the xml_index table. However, if you choose to use the xml_index the total length of the paths will still be limited to the column width of the xml_index.path column, which is currently 200 characters. So it is better for people to use the default of "usexmlindex=false".

I need someone knowledgeable with metacat to review my changes as these are pretty signficant architectural changes.

#2 - 03/31/2004 12:39 PM - Matt Jones

Now this change has been tested on postgres and oracle. A bug in the SQL that prevented it from working on oracle is now in place. I have also added GCE documents that were known to cause the problem (because they contain long paths) to the test system and they now work fine. At this point I think all problems identified in this bug are fixed and checked in.

#3 - 03/27/2013 02:17 PM - Redmine Admin

Original Bugzilla ID was 1427