

## Metacat - Bug #111

### reading large documents from metacat is slow

08/31/2000 01:04 PM - Matt Jones

<b>Status:</b>	Resolved	<b>Start date:</b>	08/31/2000
<b>Priority:</b>	Immediate	<b>Due date:</b>	
<b>Assignee:</b>	Matt Jones	<b>% Done:</b>	0%
<b>Category:</b>	metacat	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	Beta1 (AnnMeet2000)	<b>Spent time:</b>	0.00 hour
<b>Bugzilla-Id:</b>	111		

#### Description

Reading documents from metacat seems to scale with document size, and gets to be extremely slow for even medium sized documents. This is probably because we recursively create a whole tree of objects representing every element and attribute and test node when reading from the database, and this is an expensive operation. Need to change this so that the conversion from the resultset into an XML form is all done within a single instance of a class (probably within DocumentImpl.java).

#### History

##### #1 - 09/01/2000 05:16 PM - Matt Jones

Fixed document reading bug (bugzilla bug [#111](#)) so that reading documents is no longer a power function of the number of nodes in the document which used to be the case). Now, reading a document occurs entirely within DocumentImpl, by making a single SQL call to get the document data, and then using the NodeComparator class to return a TreeSet of the nodes sorted in a depth-first traversal order. This TreeSet is then processed by the new DocumentImpl.toXml() methods, which formats and outputs a text representation of the document to the Writer that is passed in. The DocumentImpl.toString() method has been re-written to utilize DocumentImpl.toXml() as well.

The old algorithm for searching (that utilized the ElementNode, textNode, CommentNode, and PINode classes) is still implemented for comparison purposes, and can be accessed by calling the readUsingSlowAlgorithm() method. A timing option has been added to DocumentImpl.main() so that the methods can be compared (see the -t and -old options). Although the difference in read time is only a fraction of a second for small documents (< 1K), the new method of reading is 72 times faster than the old method for a 34K document (1.9 seconds versus 144 seconds). This difference continues to grow as the node count increases.

##### #2 - 03/27/2013 02:13 PM - Redmine Admin

Original Bugzilla ID was 111