Metacat - Bug #4083

Metacat doesn't declare XML document encoding

05/18/2009 06:26 PM - Shaun Walbridge

Status:	Resolved	**Start date:**	05/18/2009
Priority:	Normal	**Due date:**	
Assignee:	ben leinfelder	**% Done:**	0%
Category:	metacat	**Estimated time:**	0.00 hour
Target version:	2.0.0	**Spent time:**	0.00 hour
Bugzilla-Id:	4083		

Description

When generating EML documents, Metacat doesn't include the encoding. Currently, the Registry expects its documents to be ISO-8859-1, and MetaCatServlet.java usually generates documents without an 'encoding' block, which should default the documents to UTF-8 (which may not be handled elsewhere correctly).

If possible, we should consistently use one encoding for our documents to prevent data munging issues.

History

#1 - 05/18/2009 07:05 PM - Matt Jones

This needs further discussion. Metacat itself shouldn't be changing the encoding. It should, however, properly accept documents in a variety of encodings and return them in the same encoding that they originated. Given Mike's changes in 1.9 whereby Metacat returns the exact document that we were sent (rather reconstructing it), this should now be the default. The only issues now arise from searching, where the documents need to be loaded into the relational database. Both postgres and oracle have a single encoding for each database instance (set when the db is created if I remember correctly), and if you create a DB in one encoding and then save character data from a different encoding in it, you can munge the data. Also, simply changing the encoding declaration from ISO to UTF doesn't actually solve anything -- it just misrepresents the data in the places where the encodings differ.

Finally, the metacat registry itself, which does create metadata docs, should be able to handle encodings properly by reading in multiple encodings and writing out either the same encoding or a universal default like UTF (after character conversion, of course) and setting the XML encoding declaration correspondingly.

#2 - 05/19/2009 05:08 PM - ben leinfelder

Even with storing the "original" document you have to be aware of the encoding when writing to the Metacat file system. And then also when reading from it and sending it to the client.

#3 - 05/19/2009 05:49 PM - Shaun Walbridge

Perhaps a couple of separate issues:

1. We should be explicit about encoding in the XML documents, so that parsers can read the documents without reverting to heuristics or our current assumption that documents are all UTF-8 (the default XML encoding). I think the best place to store this information is within the xml declaration block "encoding" attribute, but I could see the argument for storing it within the database, say in xml_documents. This should be handled by Metacat, but also, as Matt mentioned, by the applications which insert documents. We need some way during an insert to let Metacat know what the encoding of the data stream is, which is why setting the encoding attribute is the least intrusive way to get what we're looking for.

2. Because the database layer contains a single encoding, I'm under the impression that we have to use one encoding at some point (excepting generating additional databases), at least for the document fragments stored within the database. It would be nice to accurately be able to search on the data robustly, e.g. on UTF-8 characters such as foreign placenames, but I see this as a lower priority issue to (1). To accommodate this, we'd likely want to explicitly upconvert the document fragments to UTF-8 so we can guarantee searching works as expected for non-ASCII characters.

#4 - 10/26/2011 12:47 PM - Matt Jones

I think these encoding issues have been fixed by Ben in trunk, but we need to discuss it to see if anything further needs doing.

#5 - 10/28/2011 01:01 PM - ben leinfelder

For incoming XML files, Metacat will inspect the encoding specified in the prolog and use that if found; defaulting to UTF-8 otherwise. It will write this XML file to disk using that encoding and also read it from disk using that encoding (when someone requests the document).
When we write node values to the DB table, they will be encoded using the DB encoding (UTF-8) and any queries against those values should be compatible since all of this is coded as Java String (where the encoding has already been performed).

One remaining question for me is how a search term is encoded when it is submitted to the Metacat servlet. Once we have the String parameter, the [de]encoding has already been performed -- so that might be deep in the Tomcat layers.

At any rate, document storage and retrieval are encoding aware at this point.

#6 - 03/27/2013 02:25 PM - Redmine Admin

Original Bugzilla ID was 4083