Infrastructure - Bug #1195

CrudService.java error handling problem

2011-01-05 19:26 - Rob Nahf

Status:	Closed		**Start date:**	
Priority:	Normal		**Due date:**	
Assignee:	Robert Waltz		**% Done:**	100%
Category:	d1_common_java		**Estimated time:**	0.00 hour
Target version:	Sprint-2011.04			
Milestone:	CCI-0.5		**Story Points:**	
Product Version:	*			

Description

CrudService.create is throwing an error that cannot be parsed by the client. It appears to be that the error message itself has illegal xml characters, and they are not getting encoded before being wrapped in xml. The error statement that cannot be parsed by DocumentBuilder in d1_libclient_java's D1Node.java:

<?xml version="1.0"?>

Unexpected error in CrudService.create: Error inserting or updating document: <?xml version="1.0"?>Error running xpath expression:
//dateTimeDomain|//nonNumericDomain|//numericDomain|//access|//attributeList|//constraint|//coverage|//temporalCoverage|//geographicCoverage|//taxonomicCoverage|/dataset|/eml/dataset|//dataSource|//dataTable|//otherEntity|//citation|//address|//conferenceLocation|//party|//originator|//creator|//contact|//publisher|//editor|//recipient|//performer|//institution|//metadataProvider|//associatedParty|//personnel|//physical|//connectionDefinition|//distribution|//researchProject|//project|//relatedProject|//software|//spatialRaster|//spatialReference|//spatialVector|//storedProcedure|//view|//protocol|//additionalMetadata : The element type "datasetXXX" must be terminated by the matching end-tag "".

D1Node.java catches this as a SAXException, giving the message:
[Fatal Error] :3:99: The processing instruction target matching "[xX][mM][lL]" is not allowed.

History

#1 - 2011-01-05 19:38 - Rob Nahf

- Category set to Metacat

#2 - 2011-01-05 19:55 - Robert Waltz

- Assignee set to Robert Waltz

- Milestone set to CCI-0.5

- Category changed from Metacat to d1_common_java

- rob, 14:34 -
  the description of the error is an error xml itself, too. (That might be valid, though)

- Robert, 14:34 -
  maybe we should be grouping results in the   as CDATA

- rob, 14:34 -
  I'm writing a bug (1195)
  I was thinking the same thing.

- Robert, 14:34 -
  ok

i think it would be an easy fix. jibx does not do the error reporting
its all hardcoded

- rob, 14:35 -
  this one is from metacat. there are probably others that might present themselves elsewhere

- Robert, 14:36 -
  true enough, probably need to look through the gmn code

- vieglais, 14:37 -
  Isn't the error being serialized by jaxb for transmission?
  i mean jibx

- Robert, 14:37 -
  no
  thats the one exception to the rule

- vieglais, 14:38 -
  so, yeah they need either a CDATA block or at least proper encoding of reserved chars so output is valid xml.

- Robert, 14:39 -
  CDATA would be easiest
  just add it in the BaseException class

- vieglais, 14:39 -
  need to check the client code to see what is expected there
  for de-serializing errors

- Robert, 14:40 -
  and probably enter a tag into the html serialization too
  we do not deserialize

- vieglais, 14:42 -
  I thought the design was to deserialize and re-raise the exception on the client side?

- Robert, 14:43 -
  oh, right i think we port
  we try to build an XML document
  and based on the results of the xml document create a new exception and throw it
  so, we will need to re-write error handling on the client side
  make certain we are pulling from the correct node
  of the element

- vieglais, 14:49 -
  remind me - is it necessary to declare CDATA in the schema type definition?

- Robert, 14:49 -
  i honestly don't remember
  i think it is assumed as content of any element

- vieglais, 14:50 -
  Do we even define an Exception type in the schema?

- Robert, 14:50 -
  nope

- vieglais, 14:51 -
  so... seems like a CDATA block makes the most sense - especially if we're going to stuff XML docs inside the error description element.

- Robert, 14:51 -
  agreed

#3 - 2011-01-05 19:57 - Robert Waltz

- vieglais, 14:53 - of course, those XML docs stuffed in the CDATA block will still need to have "]]>" escaped or censored so it may be just as simple to simply "xmlescape" the error message.

#4 - 2011-01-05 20:52 - Robert Waltz

- matt, 15:43 -
  Robert: the BaseException.serializeXML() method is the problem. If we just changed that to build a DOM and then call serialize on that, the XML parser would handle the escaping for us.

- vieglais, 15:48 -
  are there unit tests on error generation?

- matt, 15:48 -
  minimal
  and certainly not on XML character encoding -- else it would have caught it
  the common lib does have a few tests though -- round trip to serialize and deserialize some errors in all 3 formats

- Robert, 15:49 -
  so part of the bug is to test this stuff, yes?

#5 - 2011-01-10 18:24 - Dave Vieglais

- Position deleted (1)

- Target version changed from Sprint-2011.01 to Sprint-2011.02

- Position set to 26

#6 - 2011-01-17 16:24 - Dave Vieglais

- Position deleted (26)

- Target version changed from Sprint-2011.02 to Sprint-2011.03

- Position set to 2

#7 - 2011-01-24 05:00 - Dave Vieglais

- Position deleted (12)

- Target version changed from Sprint-2011.03 to Sprint-2011.04

- Position set to 1

#8 - 2011-01-31 17:28 - Robert Waltz

- Status changed from New to Closed