Infrastructure - Task #1143

Bug # 1141 (Closed): unicode guid handling in mn.create()

address semi-colon issue

2010-12-03 21:33 - Rob Nahf

Status:	Closed	**Start date:**	2010-12-03
Priority:	High	**Due date:**	
Assignee:	Rob Nahf	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:			
Milestone:		**Story Points:**	
Product Version:	*		

Description

identifiers with semi-colons are failing mn.create(), giving a null-pointer exception. See parent bug.

*semi-colon used to be a security problem in tomcat 4 and 5 (http://www.rapid7.com/vulndb/lookup/http-apache-tomcat-semicolon-dir-traversal) but should be fixed.

History

#1 - 2010-12-15 19:54 - Rob Nahf

- Priority changed from Normal to High

Looks to be 1-2 problems here. From the log files, it looks like an error was by metacat regarding mismatched GUIDs. The second is an error in libclient_java, getting to an unhandled null pointer exception thrown when trying to deserialize the response from the create request.

logfiles give the following:

/var/log/apache2/other_vhosts_access.log : 0001 cn-test-1.dataone.org:80 129.24.0.10 - - [15/Dec/2010:19:23:31 +0000] "POST /knb/d1/object/knb:testid:*common-unicode-ascii-safe-;:@$-*.!*()',~_2010349111423109?sessionid=22E694DA45FD9DD06E3DCC3D\
5633BD06 HTTP/1.1" 400 389 "-" "Java/1.6.0_22"

/var/metacat/logs/knb.log : 0004 knb 20101215-19:23:31: [ERROR]: ResourceHandler: Serializing exception with code 400: GUID in method call does not match GUID in system metadata. [edu.ucsb.nceas.metacat.restservice.ResourceHandler]

The stack trace:
error message: null
java.lang.NullPointerException
at org.jibx.runtime.impl.InByteBuffer.fillBuffer(InByteBuffer.java:112)
at org.jibx.runtime.impl.InByteBuffer.require(InByteBuffer.java:206)
at org.jibx.runtime.impl.InputStreamWrapper.require(InputStreamWrapper.java:137)
at org.jibx.runtime.impl.InputStreamWrapper.getReader(InputStreamWrapper.java:239)
at org.jibx.runtime.impl.XMLPullReaderFactory$XMLPullReader.setDocument(XMLPullReaderFactory.java:217)
at org.jibx.runtime.impl.XMLPullReaderFactory$XMLPullReader.access$100(XMLPullReaderFactory.java:169)
at org.jibx.runtime.impl.XMLPullReaderFactory.recycleReader(XMLPullReaderFactory.java:151)
at org.jibx.runtime.impl.XMLPullReaderFactory.createReader(XMLPullReaderFactory.java:126)
at org.jibx.runtime.impl.UnmarshallingContext.setDocument(UnmarshallingContext.java:309)
at org.jibx.runtime.impl.UnmarshallingContext.setDocument(UnmarshallingContext.java:325)
at org.jibx.runtime.impl.UnmarshallingContext.unmarshalDocument(UnmarshallingContext.java:2899)
at org.dataone.client.D1Node.deserializeServiceType(D1Node.java:527)
at org.dataone.client.D1Node.handleCreateOrUpdate(D1Node.java:586)
at org.dataone.client.MNode.create(MNode.java:268)

#2 - 2010-12-15 19:59 - Rob Nahf

- Assignee set to Chad Berkley

#3 - 2010-12-17 22:56 - Rob Nahf

- Assignee changed from Chad Berkley to Rob Nahf

This turns out to be a problem in apache or mod_jk whereby an identifier such as "foo;bar" is truncates to "foo". Discussion about appropriateness of this with regards to FRC3986 at https://jira.springsource.org/browse/SEC-1584

Looks like we have latitude to encode the offending ";" character, and %3B does pass through all the way to metacat.

#4 - 2010-12-17 23:32 - Rob Nahf

updated the encoding methods, so now all of the identifier is getting through. However, the %3B is being decoded to ;, so an error is still occurring. Test again when the pathinfo change is made in the cn-core. (This should leave the %3B intact.)

#5 - 2010-12-20 23:10 - Chad Berkley

- Status changed from New to Closed

- % Done changed from 0 to 100

Fixed this using the tomcat filter chain. D1URLFilter now intercepts all d1/ urls coming from tomcat and handles the passes off the request as a D1HttpRequest bypassing the default getPathInfo() method so that paths with reserved characters are passed correctly.