Infrastructure - Task #1278

Story # 1275 (Closed): Fix metacat bugs

Metacat creates data files with extra new line

2011-01-28 17:41 - Chad Berkley

Status:	Closed	**Start date:**	2011-01-28
Priority:	Normal	**Due date:**	
Assignee:	Rob Nahf	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	Sprint-2011.17-Block.3		
Milestone:		**Story Points:**	
Product Version:	*		

Description

Need to figure out where this new line is getting added, probably the old MMP code. Checksums on insert should be the same as on get.

Related issues:

Related to Infrastructure - Task #828: A newline is removed from Sci metadata...	**Closed**	**2010-10-06**

History

#1 - 2011-03-07 17:32 - Dave Vieglais

- Assignee changed from Chad Berkley to Rob Nahf

#2 - 2011-03-07 18:06 - Rob Nahf

- translation missing: en.field_remaining_hours set to 8.0

#3 - 2011-03-07 22:59 - Rob Nahf

- translation missing: en.field_remaining_hours changed from 8.0 to 5.0

complicated decision tree for inserting documents into metacat. Nothing obvious in the metacat code, but there seems to be 2 conditions which lead to different routes to recording the object to metacat.

1. in MultipartRequestResolver, there's the conditional if DiskFileItem.exists(), and subsequent .write() operation if false.
2. in CrudService.create(), the conditional "isScienceMetadata(sysmeta)", leads to insertDocument() -> IOUtils.toString() on the object InputStream -> MetacatHandler.handleInsertOrUpdateAction() on true, and insertDataObject()->writeStreamToFile() -> IOUtils.copylarge() on false.

Need to find checksum conflicts in existing data.. compare sysmetadata.checksum with getChecksum().

#4 - 2011-03-10 08:14 - Rob Nahf

- File normChecksumStats.xlsx added

attaching file with checksum and object size / length information from all objects found in listObjects on cn-ucsb-1. There seem to be several inconsistencies.

#5 - 2011-03-11 17:14 - Rob Nahf

- File normChecksumStats2.xlsx added

compared 4 checksums (see attached normChecsumStats2.xlsx):

calculated from the returned object

recalculated from the returned object with appended newline ("\n")

checksum in objectList.objectInfo

checksum returned from getCheckSum call

inconsistencies:

newline missing from getObject return - recalculated checksum (2) matches those returned in objectList (3) and getChecksum (4). [CROG = "ABBB", 4 objects]

getCheckSum equals calculated, but difference between objectInfo.size and calculated size is 1 - potentially a hidden newline problem. [CROG = "ABCA", 492 objects, all of objectFormat application/octect-stream, and pid from dryad like "hdl:10255/dryad.115/mets.xml" (not of these dryad ids]

calcuated and recalculated checksums don't match objectInfo and getChecksum ones. [CROG = "ABCC", some from zero length objects returns]

#6 - 2011-03-11 23:20 - Rob Nahf

I am assuming newline = \n, and not \r, but ambiguity of how newline "token" gets converted to bytes (based on filesystem) may be contributing to the overall issue (see #828 and http://en.wikipedia.org/wiki/Newline)

#7 - 2011-03-12 17:56 - Matthew Jones

Rob -- this may already be obvious to you, but I thought I'd throw it out just in case. Rather than getting the checksums to match, you might have more luck doing a binary diff on the objects whose checksums differ. I'd use od (octal-dump) to show the actual binary values in the file, then use diff. Something like this might work to show exactly which characters differ:

```
snow:temp jones$ od -x file1.csv > od1
snow:temp jones$ od -x file2.csv > od2
snow:temp jones$ diff -u od1 od2
--- od1 2011-03-12 08:51:31.000000000 -0900
+++ od2 2011-03-12 08:52:15.000000000 -0900
@@ -1,2 +1,2 @@
-0000000      2c31    2c32    0a33    2c34    2c35    0a36

+0000000      2c31    2c32    0a33    2c34    2c35    0036

0000014
```

You also may just want to use od to see then end of the files and see if the final chars are different newlines, or end in two newlines, etc.
od -cx file1.csv is a convenient way to see non-printing characters.

#8 - 2011-03-14 18:41 - Rob Nahf

- % Done changed from 0 to 70

- translation missing: en.field_remaining_hours changed from 5.0 to 8.0

laptop failure caused complete loss of code. will need to redo some of the work.

#9 - 2011-03-25 20:56 - Rob Nahf

Eureka! Better client tests for checksum show that the last trailing LF or CRLF are not returned upon object retrievals.

Notice that checksums in these simple cases return the checksum of objects without concatenated line separators.
Double sequences have polymorphic behavior: \n\n becomes \n, yet \r\n\r\n also becomes \n (neither \r is preserved)

********** running test for testChecksum. node = http://cn-dev.dataone.org/knb/d1/ **********

Test file = /d1_testdocs/checksumTestSet/sciMD-eml-201-NoLastLForCR.xml
File concatenations = nothing to concatenate
Checksum 1: D3339CAF019C3EB40811CC191DCDEB12
getChecksum value: D3339CAF019C3EB40811CC191DCDEB12
retrieved object checksum: D3339CAF019C3EB40811CC191DCDEB12

Test file = /d1_testdocs/checksumTestSet/sciMD-eml-201-NoLastLForCR.xml
File concatenations = 0x0A
Checksum 1: 4FAE634193828BCCC02F41E0C51AE839
getChecksum value: 4FAE634193828BCCC02F41E0C51AE839
retrieved object checksum: D3339CAF019C3EB40811CC191DCDEB12

Test file = /d1_testdocs/checksumTestSet/sciMD-eml-201-NoLastLForCR.xml
File concatenations = 0x0D 0x0A
Checksum 1: A3B0DEA32B5A2F952A49F715699DCF8B
getChecksum value: A3B0DEA32B5A2F952A49F715699DCF8B
retrieved object checksum: D3339CAF019C3EB40811CC191DCDEB12

Test file = /d1_testdocs/checksumTestSet/sciMD-eml-201-NoLastLForCR.xml
File concatenations = 0x0A 0x0A
Checksum 1: 2AFF44C89AFF7524455502065D795543
getChecksum value: 2AFF44C89AFF7524455502065D795543
retrieved object checksum: 4FAE634193828BCCC02F41E0C51AE839

Test file = /d1_testdocs/checksumTestSet/sciMD-eml-201-NoLastLForCR.xml
File concatenations = 0x0D 0x0A 0x0D 0x0A
Checksum 1: D72BD1F476091C353F267EE908BEEBB9
getChecksum value: D72BD1F476091C353F267EE908BEEBB9
retrieved object checksum: 4FAE634193828BCCC02F41E0C51AE839

#10 - 2011-03-28 22:11 - Rob Nahf

the create documents on cn-dev have the appropriate \n or \r\n, so the initial write operation successfully preserves the file contents.

The problem has been isolated to .edu.ucsb.nceas.utilities.FileUtil, where a BufferedReader is used to read the stored file from the filesystem, remove incompatible content (based on objectFormat) for the client, and write to the output stream. Current implementation uses the readline method to read from the file, and reassembles the lines assuming a newline was the original separator, and that the file does have a line separator(s) at the end of the file.

readline behavior from the javadoc:

Reads a line of text. A line is considered to be terminated by any one of a line feed ('\n'), a carriage return ('\r'), or a carriage return followed immediately by a linefeed.
Returns:
A String containing the contents of the line, not including any line-termination characters, or null if the end of the stream has been reached

This is insufficient for preserving original content because (a) there's no way to determine the original line separator and (b) there's no way to tell if the last line contained a line separator or not.

#11 - 2011-04-01 19:56 - Rob Nahf

- % Done changed from 70 to 100

- Status changed from In Progress to Closed

committed changes to utilities project FileUtil class, and tagged the version with this fix as UTILITIES_1_1_0_RC2.

#12 - 2011-04-01 20:36 - Rob Nahf

checked DocumentImpl for newline addition mentioned in task #828. it seems to have been already removed. So don't think there are any loose ends...

#13 - 2011-04-25 16:23 - Rob Nahf

- translation missing: en.field_remaining_hours changed from 8.0 to 0.0

Files

normChecksumStats.xlsx	282 KB	2011-03-10	Rob Nahf
normChecksumStats2.xlsx	567 KB	2011-03-11	Rob Nahf