Archivematica - Bug #9220

Characterize and extract metadata hangs when ffprobe writes non-UTF-8 character to stderr

12/07/2015 06:40 PM - Andrew Berger

Status:	New	**Start date:**	12/07/2015
Priority:	Medium	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
Google Code Legacy ID:		**Pull Request:**	
Sponsored:	No	**Requires documentation:**	

Description

I've noticed that for certain MOV files, the characterize and extract metadata microservice hangs without completing or failing (see attached screenshot). After investigating the issue, I believe this is the result of FFprobe writing an error message that contains a character that is not valid UTF-8. Although Archivematica does not fail when trying to process these files, the MCPServer.debug.log does show the following error (see attached file for fuller log):

ERROR 2015-12-07 22:01:04 archivematica.mcp.server:utils:wrapped:16: Uncaught exception
Traceback (most recent call last):
File "/usr/lib/archivematica/MCPServer/utils.py", line 14, in wrapped
return fn(*args, **kwargs)
File "/usr/lib/archivematica/MCPServer/taskStandard.py", line 86, in performTask
self.check_request_status(completed_job_request)
File "/usr/lib/archivematica/MCPServer/taskStandard.py", line 95, in check_request_status
self.linkTaskManager.taskCompletedCallBackFunction(self)
File "/usr/lib/archivematica/MCPServer/linkTaskManagerFiles.py", line 144, in taskCompletedCallBackFunction
databaseFunctions.logTaskCompletedSQL(task)
File "/usr/lib/archivematica/archivematicaCommon/databaseFunctions.py", line 236, in logTaskCompletedSQL
task.save()
File "/usr/local/lib/python2.7/dist-packages/django/db/models/base.py", line 546, in save
force_update=force_update, update_fields=update_fields)
File "/usr/local/lib/python2.7/dist-packages/django/db/models/base.py", line 626, in save_base
rows = manager.using(using).filter(pk=pk_val)._update(values)
File "/usr/local/lib/python2.7/dist-packages/django/db/models/query.py", line 605, in _update
return query.get_compiler(self.db).execute_sql(None)
File "/usr/local/lib/python2.7/dist-packages/django/db/models/sql/compiler.py", line 1014, in execute_sql
cursor = super(SQLUpdateCompiler, self).execute_sql(result_type)
File "/usr/local/lib/python2.7/dist-packages/django/db/models/sql/compiler.py", line 840, in execute_sql
cursor.execute(sql, params)
File "/usr/local/lib/python2.7/dist-packages/django/db/backends/mysql/base.py", line 120, in execute
return self.cursor.execute(query, args)
File "/usr/lib/python2.7/dist-packages/MySQLdb/cursors.py", line 176, in execute
if not self._defer_warnings: self._warning_check()
File "/usr/lib/python2.7/dist-packages/MySQLdb/cursors.py", line 92, in _warning_check
warn(w[-1], self.Warning, 3)
Warning: Invalid utf8 character string: 'A96461'

The invalid character string appears to be coming from the following lines in FFprobe's error output:

[mov,mp4,m4a,3gp,3g2,mj2 0x2e9e8c0] overread end of atom '<U+FFFD>day' by 4 bytes
[mov,mp4,m4a,3gp,3g2,mj2 0x2e9e8c0] overread end of atom '<U+FFFD>swr' by 4 bytes

With the text encoding set to UTF-8, the first character in the two atom names does not display properly in my system. I believe this is because that character is written as '0xa9' in the FFmpeg code itself, where it's the first character of various different atom names:

https://github.com/FFmpeg/FFmpeg/blob/82c5f3178930285f84c42ab4b026ee48d53305ec/libavformat/mov.c#L328

As far as I can tell, the character is supposed to represent a copyright symbol and is probably meant to be displayed in an encoding

like ISO-8859-1. It would need to be converted to be proper UTF-8.

As a workaround, since we have dozens of files to ingest that will generate some form of this error message, I've modified the FFprobe command in our testing FPR to the following, which converts the FFprobe stderr to utf-8 while keeping stdout the same:

{ ffprobe -i "%fileFullName%" -show_data -show_format -show_error -show_streams -show_chapters -show_private_data -show_versions -print_format xml 2>&1 1>&3 | iconv -f iso_8859-1 -t utf8 1>&2; } 3>&1

This is based on the answer here: http://unix.stackexchange.com/questions/3514/how-to-grep-standard-error-stream-stderr

In testing, this seems to have fixed the issue. FFprobe output is still written to the METS file and the error message appears as UTF-8 in the dashboard when you click on the "gear" icon to view details.

Files

invalid-utf8.log	31.9 KB	12/07/2015	Andrew Berger
characterize-extract-hangs.png	19.4 KB	12/07/2015	Andrew Berger