

Kepler - Bug #4859

GDALTranslate does not work on Windows 7 (64bit)

03/02/2010 07:41 AM - Tom Parris

Status:	In Progress	Start date:	03/02/2010
Priority:	Normal	Due date:	
Assignee:	Chad Berkley	% Done:	0%
Category:	actors	Estimated time:	0.00 hour
Target version:	2.X.Y	Spent time:	0.00 hour
Bugzilla-Id:	4859		
Description			
GDALJniGlue fails to initialize on Windows 64bit operating systems.			
ptolemy.kernel.util.IllegalArgumentException: in .GDAL_Translate_test_oak.manager Because: Could not initialize class org.ecoinformatics.seek.gis.gdal.GDALJniGlue at ptolemy.actor.Manager.execute(Manager.java:472) at ptolemy.actor.Manager.run(Manager.java:1119) at ptolemy.actor.Manager\$3.run(Manager.java:1160) Caused by: java.lang.NoClassDefFoundError: Could not initialize class org.ecoinformatics.seek.gis.gdal.GDALJniGlue at org.ecoinformatics.seek.gis.gdal.GDALTranslateActor.fire(GDALTranslateActor.java:207) at ptolemy.actor.AtomicActor.iterate(AtomicActor.java:469) at ptolemy.actor.sched.StaticSchedulingDirector.fire(StaticSchedulingDirector.java:188) at ptolemy.actor.CompositeActor.fire(CompositeActor.java:458) at ptolemy.actor.Manager.iterate(Manager.java:714) at ptolemy.actor.Manager.execute(Manager.java:349) ... 2 more Caused by: java.lang.NoClassDefFoundError: Could not initialize class org.ecoinformatics.seek.gis.gdal.GDALJniGlue at org.ecoinformatics.seek.gis.gdal.GDALTranslateActor.fire(GDALTranslateActor.java:207) at ptolemy.actor.AtomicActor.iterate(AtomicActor.java:469) at ptolemy.actor.sched.StaticSchedulingDirector.fire(StaticSchedulingDirector.java:188) at ptolemy.actor.CompositeActor.fire(CompositeActor.java:458) at ptolemy.actor.Manager.iterate(Manager.java:714) at ptolemy.actor.Manager.execute(Manager.java:349) at ptolemy.actor.Manager.run(Manager.java:1119) at ptolemy.actor.Manager\$3.run(Manager.java:1160)			

History

#1 - 03/02/2010 08:26 AM - Matt Jones

GDAL probably just needs to be recompiled for this architecture, and the installer adapted to detect which architecture is being used and provide the proper binary. We should do two things to close this bug:

1) Check all native library actors and recompile as needed for the following platforms, and recompile/install as needed:

- a) Win 32
- b) Win 64
- c) Mac OS X 32
- d) Mac OS X 64
- e) Linux 32
- f) Linux 64

2) Develop a test suite for the NMI build that will run these actors on all 6 architectures and test that they continue to function.

#2 - 03/04/2010 09:34 AM - Chad Berkley

1) Check all native library actors and recompile as needed for the following platforms, and recompile/install as needed:

- a) Win 32
- b) Win 64

Does someone have a win64 machine they could use to do this? I can provide compilation instructions.

- c) Mac OS X 32
- d) Mac OS X 64
- e) Linux 32
- f) Linux 64

I can handle these.

- 2) Develop a test suite for the NMI build that will run these actors on all 6 architectures and test that they continue to function.

This will be hard because NMI continues to have issues with many of their test platforms. Right now, I can only get things to run on one platform (linux). The windows and OSX platforms are continually over used and jobs on them end up taking forever to run. I can work on this again after the 2.0 release.

#3 - 03/04/2010 11:59 AM - Christopher Brooks

The Ptolemy Matlab interface has similar problems, I'm looking at building 64bit libraries for the Mac. On the Mac, one can have universal libraries that support both 32 and 64 bit (and presumably x86 and ppc). On the Mac, the lipo command is what is used to build universal libraries.

#4 - 03/04/2010 09:37 PM - Christopher Brooks

I created a universal binary that supports 32 and 64 bit JVMs under MacOSX. The library may be found in ptII/lib/matlabMacOSX.jar

Note that to run the 64bit version, I had to set
export

```
DYLD_LIBRARY_PATH=/Applications/MATLAB_R2009b.app/bin/maci:/Applications/MATLAB_R2009b.app/bin/maci64:/Users/cxh/ptII/lib/Users/cxh/ptII/lib:/Applications/MATLAB_R2009b.app/sys/os/maci:/Applications/MATLAB_R2009b.app/sys/os/maci64
```

so that libXm.3.so and the 64bit matlab libraries are found.

To build the 64 bit library, I used the -m64 gcc option.
To build the universal binary, I used lipo:

```
lipo -create -arch i386 libptmatlab32.jnilib -arch x86_64 libptmatlab64.jnilib -o libptmatlab.jnilib
```

Note that for Web Start to work under Mac OS 10.5 with Java 1.5.0_22, the shared libraries must have a .jnilib extension. I could not get Web Start to work if the shared libraries have a .dylib extension. It would probably be worth renaming the MacOSX shared libraries that are to be loaded using JNI to .jnilib

See

<http://developer.apple.com/Mac/library/documentation/Java/Conceptual/Java14Development/05-CoreJavaAPIs/CoreJavaAPIs.html>

I still need to create 32 and 64 bit Ptolemy/Matlab libraries for Windows and Linux. I don't think I have access to a 64 bit Windows machine though.

I still need to fix up \$PTII/lbnl, which contains the Building Controls Virtual Test Bench, which provides a socket-based interface to Matlab, C and other programs.

#5 - 03/05/2010 12:29 PM - Chad Berkley

GDAL requires Microsoft Visual C++ to compile:

<http://trac.osgeo.org/gdal/wiki/GdalOgrInJavaBuildInstructions>

I don't have a copy of this, nor do I have a 64 bit windows machine. Is there anyone out there that has this combo? If not, I don't see a way to compile gdal for win64.

#6 - 03/08/2010 05:45 AM - Tom Parris

Apologies for the delay. I've been travelling.

We obviously have the requisite hardware and can compile the GDAL C libraries. But we need some instructions on how to package things appropriately. Can someone send a recipe?

We are separately trying to compile the new GDAL JAVA language bindings to build some new actors. Stay tuned for progress on that. Assuming we are successful, this would eliminate the need for the Kepler provided JNI interface that only supports a couple of the GDAL methods.

#7 - 03/08/2010 11:20 AM - Chad Berkley

Hi Tom, The instructions for building gdal in windows are here:
<http://trac.osgeo.org/gdal/wiki/GdalOgrInJavaBuildInstructions>

We basically just need the 64 bit dll files to include with Kepler. I haven't built it since the first time I did so which was probably 5 years ago so I'm not exactly sure what that entails anymore. Let me know if you need more info and I can try to dig it up.

#8 - 03/08/2010 11:36 AM - Tom Parris

OK. We're in the process of building the new suite of java language bindings now. We've gotten them to compile on a 32 bit platform and our next step is to port to the 64bit platform.

We are doing this in preparation of writing some new Kepler GDAL actors. The new GDAL actors will read images into matrix tokens so we can create pipelines of actors that act on the data without having to read/write it for each operation. (Please let me know if I've missed something and this capability already exists elsewhere in Kepler).

I've poked around the current suite of Kepler GDAL actors. They make use of a custom coded JNI (not the generic java language bindings). So, I think we just need to provide the GDAL dll's for each platform and the dll for the JNI for the 64bit platform.

Our first priority is getting our new actors up and running on the 64bit platform. We'll try to get the existing GDAL actors up and running on 64bits after that.

-- Tom

#9 - 03/08/2010 12:42 PM - Chad Berkley

Sounds good. Thanks, Tom. Would you consider creating a new GDAL module for kepler? If so, we would be willing to host it in our SVN repository.

#10 - 03/08/2010 01:10 PM - Tom Parris

Yes. But for the moment it has to take second place to meeting our own rather extreme schedule commitments. I think we can have something in deliverable shape by late April/early May.

#11 - 03/09/2010 04:05 PM - Chad Berkley

Pushing this to post 2.0 since I think it affects a small number of users. I don't think this should hold up the 2.0 release.

#12 - 03/27/2013 02:28 PM - Redmine Admin

Original Bugzilla ID was 4859