Subject: Re: IDL Python Bridge
Posted by ivolabbe on Wed, 11 Jan 2017 22:34:39 GMT
View Forum Message <> Reply to Message

Thanks for the tip! I had started toying around with install_name_tool getting pretty far, but the instructions provided by Chris helped me figure out the rest. Chris's solution (putting the missing path information in the libraries) is the right thing to do, but not a full solution.

For example, even when setting all paths in the libraries, the IDL-Python bridge still seems to rely critically on defining PYTHONHOME in ./profile, which breaks environment switching in Conda (e.g., between 2.7 and 3.4). Setting these paths at all should not be necessary as both IDL and Python can resolve all necessary relative paths just fine upon invocation. Gratuitous use of global environment vars only lead to ambiguities and headaches ;-) I hope any future IDL-Python fix will dispense with them entirely.

In any case, for completeness I am appending the full fix below, including a workaround for the required PYTHONHOME which enables context switching between 2.7 and 3.4 in Conda. It all seems to work; the Python bridge is really great!

Ivo

```
#
# fixes IDL Python Bridge on IDL 8.5.1 on Mac OS X El Capitan 10.11.6
# for python 2.7 and 3.4 including Conda environment switching
#
# based on install_name_tool solution provided by Chris Torrence
#  http://stackoverflow.com/questions/38468764/idl-python-bridg
e-fails-due-to-sip-on-osx-el-capitan
#

# add these to .profile :
export PATH=/Users/username/anaconda/bin:${PATH}
export PYTHONHOME=/Users/username/anaconda
export  PYTHONPATH=/Users/username/Applications/exelis/idl/bin/bin.d arwin.x86_64
export  PYTHONPATH=${PYTHONPATH}:/Users/username/Applications/exelis /idl85/lib/bridges

# note: this (non-standard) use of PYTHONPATH/PYTHON is incompatible with
# Conda environment switching. An ideal solution would not require any global
# paths at all.  Workaround: reset PYTHONHOME after switching
alias py34='source activate py34 ; export PYTHONHOME=$CONDA_PREFIX'
alias py27='source activate py27 ; export PYTHONHOME=$CONDA_PREFIX'

# Hopefully future IDL-Python bridge fixes will dispense with
# global environment variables. Less ambiguity = better.
#
# Python will default to the directory where the executable was found
# (and relative paths).  Since the IDL-Python Bridge upon invocation
```

```
# by definition knows the location of both executables, it should be
# able to resolve all necessary default idl and python libraries
# without any additional help.

# admin privileges needed for global fix
# but not for user level install in ~/Applications
cd /Applications/exelis/idl85/bin/bin.darwin.x86_64/
sudo install_name_tool -change libpython3.4m.dylib
/Users/username/anaconda/envs/py34/lib/libpython3.4m.dylib idl_python34.so
sudo install_name_tool -change libpython2.7.dylib
/Users/username/anaconda/lib/libpython2.7.dylib idl_python27.so

# verify with
otool -L idl_python34.so
otool -L idl_python27.so

sudo install_name_tool -change libidl.8.5.dylib @loader_path/libidl.8.5.dylib pythonidl34.so
sudo install_name_tool -change libidl_ips.8.5.dylib @loader_path/libidl_ips.8.5.dylib
pythonidl34.so
sudo install_name_tool -change libpython3.4m.dylib
/Users/username/anaconda/envs/py34/lib/libpython3.4m.dylib pythonidl34.so

sudo install_name_tool -change libidl.8.5.dylib @loader_path/libidl.8.5.dylib pythonidl27.so
sudo install_name_tool -change libidl_ips.8.5.dylib @loader_path/libidl_ips.8.5.dylib
pythonidl27.so
sudo install_name_tool -change libpython2.7.dylib
/Users/username/anaconda/lib/libpython2.7.dylib pythonidl27.so

sudo install_name_tool -change libMesaGL6_2.dylib @loader_path/libMesaGL6_2.dylib
libidl.8.5.dylib
sudo install_name_tool -change libMesaGLU6_2.dylib @loader_path/libMesaGLU6_2.dylib
libidl.8.5.dylib
sudo install_name_tool -change libOSMesa6_2.dylib @loader_path/libOSMesa6_2.dylib
libidl.8.5.dylib
sudo install_name_tool -change libXm.3.0.2.dylib @loader_path/libXm.3.0.2.dylib libidl.8.5.dylib
sudo install_name_tool -change libMesaGL6_2.dylib @loader_path/libMesaGL6_2.dylib
libMesaGLU6_2.dylib
sudo install_name_tool -change libMesaGL6_2.dylib @loader_path/libMesaGL6_2.dylib
libOSMesa6_2.dylib
sudo install_name_tool -change libidl.8.5.dylib @loader_path/libidl.8.5.dylib libidl_ips.8.5.dylib


# tests without workaround for environment switching
# running without PYTHONHOME set

IDL>  ran = Python.Import('numpy.random')
% Loaded DLM: PYTHON34.
Could not find platform independent libraries <prefix>
```

Could not find platform dependent libraries <exec_prefix>
Consider setting $PYTHONHOME to <prefix>[:<exec_prefix>]
Fatal Python error: Py_Initialize: unable to load the file system codec
ImportError: No module named 'encodings'
Current thread 0x00007fff76ca3000 (most recent call first):
Abort trap: 6

with PYTHONHOME IDL-Python works great, but Conda breaks when
when switching environments:

(py27) $ source activate py34
(py34) $ python
Fatal Python error: Py_Initialize: unable to load the file system codec
ImportError: No module named 'encodings'
Current thread 0x00007fff76ca3000 (most recent call first):
Abort trap: 6
ujamaa:~ username$ python -v
import _frozen_importlib # frozen
import imp # builtin
import sys # builtin
# installing zipimport hook
# installed zipimport hook
Fatal Python error: Py_Initialize: unable to load the file system codec
Traceback (most recent call last):
  File "<frozen importlib._bootstrap>", line 2237, in _find_and_load
  File "<frozen importlib._bootstrap>", line 2224, in _find_and_load_unlocked
ImportError: No module named 'encodings'
Abort trap: 6

with the current fix:

(py27) $ idl
IDL Version 8.5.1, Mac OS X (darwin x86_64 m64).
(c) 2015, Exelis Visual Information Solutions, Inc., a subsidiary of Harris Corporation.
IDL> >>>
>>> import sys
% Loaded DLM: PYTHON27.
>>> sys.version
'2.7.13 |Anaconda 2.3.0 (x86_64)| (default, Dec 20 2016, 23:05:08) \n[GCC 4.2.1 Compatible
Apple LLVM 6.0 (clang-600.0.57)]'
>>>
IDL> exit
(py27) $ py34
(py34) $ idl
IDL Version 8.5.1, Mac OS X (darwin x86_64 m64).
(c) 2015, Exelis Visual Information Solutions, Inc., a subsidiary of Harris Corporation.
IDL> >>>

```
>>>  import sys
% Loaded DLM: PYTHON34.
>>>  sys.version
'3.4.4 |Continuum Analytics, Inc.| (default, Jun 15 2016, 15:48:32) \n[GCC 4.2.1 Compatible Apple
LLVM 4.2 (clang-425.0.28)]'
>>>
IDL> exit

Great success!!




;
; more infoe
;
;
; decription of problem

; Some IDL 8.5.1 libraries have incomplete or missing path information
; due to omissions during compiling and linking. These missing paths
; break the IDL-Python bridge on Mac OS X Capitan on IDL 8.5.1
; (the issue is rumored to have been fixed in 8.6, but I don't know
; the details, i.e. whether it is a fix or work around).
;
; On some platforms these missing paths are "fixed" by setting various
; (DY)LD_LIBRARY search paths, so that the libraries are found anyway
; but this is essentially crude hack which can have unintended
; consequences on system libraries. In addition, the upgraded
; System Integrity Protection (SIP) of El Capitan limits the usage
; of DYLD_LIBRARY_PATH even further, making this workaround impossible.
; IDL/Python is not unique in this; the internet forums are alight
; with developers panicking over broken links.
;
; A solution:
;
; Rather than toying around with system paths and potentially breaking
; other things, the libraries should simply be fixed by receiving the
; correct (and sufficient) path information. On Mac OS X this can be
; achieved with the install_name_tool program. We thus attempt to locate
; all libraries and then call 'sudo install_name_tool' to add path
; information to them. Obviously, the root problem is best addressed at
; compile time by the maintainers of IDL.
;
; Some background info:
; - similar problem:  http://stackoverflow.com/questions/23771608/trouble-installi
ng-galsim-on-osx-with-anaconda
; - practical tutorial on otool+install_name_tool:
```

http://matthew-brett.github.io/docosx/mac_runtime_link.html
; - concise overview of rpath etc:
https://wincent.com/wiki/@executable_path,_@load_path_and_@r path
; - http://conda.pydata.org/docs/building/shared-libraries.html# shared-libraries-in-linux-and-os-x
;

On Wednesday, 11 January 2017 05:33:04 UTC+1, Jim P wrote:
> On Tuesday, January 10, 2017 at 3:43:19 PM UTC-7, ivol...@gmail.com wrote:
>> On Tuesday, 21 June 2016 22:29:13 UTC+2, guy...@gmail.com wrote:
>>> I eventually got this to work using the "install_name" tool with the "-change" flag to fix the dylib which otherwise caused failure despite proper environment variables. This solved my problem for both Python 3.4.4 and Python 2.7.11
>>
>> I seem to be running into the same problem with a failing IDL-Python bridge.
>>
>> It appears to be a well understood issue related to DYLD_LIBRARY_PATH on Mac OS X Caption, and it is reported solved in 8.6 (http://www.harrisgeospatial.com/Home/NewsUpdates/TabId/170/ArtMID/735/ArticleID/14961/IDL-86-Release-Notes.aspx IDL-69603).
>>
>> However, I can not find a solution or workaround for 8.5.1 anywhere.
>>
>> Would you mind sharing your install_name_tool fix ? I'm quite eager to get it to work!
>>
>> thanks
>
> Do the instructions Chris provided in the stackoverflow.com discussion related to this topic help? Search for the discussion titled "IDL-Python bridge fails due to SIP on OSX El Capitan".
>
> Jim P.