Subject: IDL Python Bridge
Posted by Phillip Bitzer on Thu, 04 Feb 2016 19:47:13 GMT
View Forum Message <> Reply to Message

Before I chase the rabbit down the hole, I'm hoping someone else has some experience in getting this to work.

I am unable to get the IDL-Python bridge up and running on a Mac (OS 10.9.5, for completeness). I am using Anaconda and IDL 8.5. Let me check my Python install, as in the Exelis, er Harris, installation directions (https://www.exelisvis.com/docs/python.html and http://www.exelisvis.com/Support/HelpArticlesDetail/TabId/21 9/ArtMID/900/ArticleID/14600/Using-the-IDL-Python-Bridge-on- Mac.aspx):

```
>>>  import sys
>>>  sys.version
'2.7.11 |Anaconda 2.4.1 (x86_64)| (default, Dec  6 2015, 18:57:58) \n[GCC 4.2.1 (Apple Inc. build 5577)]'
```

Before setting any variables, I'll check my python version at the command line too:

```
$ python --version
Python 2.7.11 :: Anaconda 2.4.1 (x86_64)
```

and:

```
$ which python
~/anaconda/bin/python
```

And for kicks and giggles, try the bridge anyway:

```
IDL> >>>
>>>  import sys
% Loaded DLM: PYTHON27.
ImportError: numpy.core.multiarray failed to import
Segmentation fault: 11
```

and I get dumped back to the command line.

Not surprising it doesn't work - I didn't follow the installation directions. So, let me set the paths as specified. I working with a bash shell, so I'll be exporting. No big deal. (One caveat, it seems Anaconda has already added itself to my PATH, so no need to do it again.)

Source my bash profile, echo all the paths, and everything seems to be there. So, let's try the bridge again:

```
IDL> >>>
>>>  import sys
% Loaded DLM: PYTHON27.
ImportError: numpy.core.multiarray failed to import
Segmentation fault: 11
```

and back to the command line. Again.

Hmm. Let's check that version now:

```
python --version
Python 2.7.5
```

Er, it seems I've pointed away from the Anaconda python. But,

```
$ which python
~/anaconda/bin/python
```

is still the same. Sigh.

I have spend some time digging around, and it seems it has something to with the DYLD environment variable, most likely the /usr/bin part. It seems I'm picking up the python in /usr/bin instead of the Anaconda version.

Clearly, this is a path problem - which I know is a PITA to troubleshoot remotely.

All this ask: does anyone have the bridge up and running on a Mac? If so, are your paths any different? I can't seem to track down why python is now pointing at an older install. This seems to be the issue causing the IDL-Python bridge fits.

Subject: Re: IDL Python Bridge
Posted by Michael Galloy on Fri, 05 Feb 2016 03:43:05 GMT

View Forum Message <> Reply to Message

On 2/4/16 12:47 pm, Phillip Bitzer wrote:

> All this ask: does anyone have the bridge up and running on a Mac? If
> so, are your paths any different? I can't seem to track down why
> python is now pointing at an older install. This seems to be the
> issue causing the IDL-Python bridge fits.

It doesn't work for me with OS X El Capitan and IDL 8.5.1 right now,
though it was working for me this summer when IDL 8.5 came out.

Mike
--
Michael Galloy
www.michaelgalloy.com
Modern IDL: A Guide to IDL Programming (http://modernidl.idldev.com)

Subject: Re: IDL Python Bridge
Posted by lecacheux.alain on Fri, 05 Feb 2016 08:57:36 GMT

View Forum Message <> Reply to Message

Le vendredi 5 février 2016 04:43:11 UTC+1, Mike Galloy a écrit :
> On 2/4/16 12:47 pm, Phillip Bitzer wrote:
>> All this ask: does anyone have the bridge up and running on a Mac? If
>> so, are your paths any different? I can't seem to track down why
>> python is now pointing at an older install. This seems to be the
>> issue causing the IDL-Python bridge fits.
>
> It doesn't work for me with OS X El Capitan and IDL 8.5.1 right now,
> though it was working for me this summer when IDL 8.5 came out.
>
> Mike
> --
> Michael Galloy
> www.michaelgalloy.com
> Modern IDL: A Guide to IDL Programming (http://modernidl.idldev.com)

For info, Python bridge works very well both in IDL 8.5 and 8.5.1, while using Anaconda Python 3.4 and MS-Windows 64 bits (+ seamless install !).
alx.

Subject: Re: IDL Python Bridge
Posted by vkashyap on Thu, 16 Jun 2016 15:20:51 GMT
View Forum Message <> Reply to Message

Am trying to set up and run it on Yosemite, and encountering very similar problems. No path problems, but errors of the sort numpy.core.multiarray failed to import, and a segfault upon
>>>import sys

Happens regardless of whether I setenv IDL_DLM_PATH \<IDL-DEFAULT\> or not.
DYLD_LIBRARY_PATH, PYTHONHOME, and PYTHONPATH have been set as recommended.

IDL 8.5
Mac OS X 10.10.5 Yosemite
anaconda python 2.7.10

VK

Subject: Re: IDL Python Bridge
Posted by guygma on Thu, 16 Jun 2016 19:48:58 GMT
View Forum Message <> Reply to Message

Similar issues here on OSX10.11. Segfault upon import of "sys" module while in a 2.7.11 virtualenv (using pyenv installed from homebrew). Import with either 3.4.4 or 3.5.1 version of python give a different error which makes me think that the bridge is doing something more convoluted with the path than Exelis is letting on...

From the python side, importing idlpy fails with an ImportError:

```
Python 3.4.4 (default, Jun 16 2016, 15:18:26)
[GCC 4.2.1 Compatible Apple LLVM 7.3.0 (clang-703.0.31)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>  import idlpy
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/Applications/exelis/idl/lib/bridges/idlpy.py", line 27, in <module>
    raise(e)
  File "/Applications/exelis/idl/lib/bridges/idlpy.py", line 21, in <module>
    pyidl = importlib.import_module(idllibrary)
  File " /usr/local/var/pyenv/versions/3.4.4/Python.framework/Version
s/3.4/lib/python3.4/importlib/__init__.py ", line 109, in import_module
    return _bootstrap._gcd_import(name[level:], package, level)
ImportError:  dlopen(/Applications/exelis/idl/bin/bin.darwin.x86_64/python idl34.so, 2): Library not
loaded: libidl_ips.8.5.dylib
  Referenced from:  /Applications/exelis/idl/bin/bin.darwin.x86_64/pythonidl34.s o
  Reason: image not found
>>>  import sys
>>>  sys.path
['', '/Users/Guy/tmp', '/Applications/exelis/idl/bin/bin.darwin.x86_64',
'/Applications/exelis/idl/lib/bridges',  '/usr/local/var/pyenv/versions/3.4.4/Python.framework/Versio
ns/3.4/lib/python34.zip',  '/usr/local/var/pyenv/versions/3.4.4/Python.framework/Versio
ns/3.4/lib/python3.4',  '/usr/local/var/pyenv/versions/3.4.4/Python.framework/Versio
ns/3.4/lib/python3.4/plat-darwin',  '/usr/local/var/pyenv/versions/3.4.4/lib/python3.4/lib-dynlo ad',
'/usr/local/var/pyenv/versions/3.4.4/Python.framework/Versio ns/3.4/lib/python3.4/site-packages',
'/usr/local/var/pyenv/versions/3.4.4/lib/python3.4/site-pack ages']
```

Subject: Re: IDL Python Bridge
Posted by guygma on Tue, 21 Jun 2016 20:29:09 GMT

View Forum Message <> Reply to Message

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

Subject: Re: IDL Python Bridge
Posted by ivolabbe on Tue, 10 Jan 2017 22:43:17 GMT

View Forum Message <> Reply to Message

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/A
rtMID/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

Subject: Re: IDL Python Bridge
Posted by Jim Pendleton on Wed, 11 Jan 2017 04:33:02 GMT

View Forum Message <> Reply to Message

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/A
rtMID/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.

Subject: Re: IDL Python Bridge
Posted by ivolabbe on Wed, 11 Jan 2017 22:34:39 GMT

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.

Subject: Re: IDL Python Bridge
Posted by ivolabbe on Sun, 29 Jan 2017 00:42:30 GMT

View Forum Message <> Reply to Message

After having experimented a while with the IDL-Python bridge (which is fantastic) on IDL 8.5.1 on Mac OS X 10.11, just a quick update to correct my previous post.

To confirm: the install_name_tool fixes of the IDL-Python bridge libraries work great for IDL 8.5.1 on Mac os X El Capitan 10.11 for both Python 2.7 and 3.4.

However, environment switching in Anaconda (4.3.7 osx-64) remained a problem. The key issue is IDL-Pythons reliance on a explicitly defined PYTHONHOME, which breaks Anaconda's environment switching.

In the end, I wrapped the call to idl, so that IDL alone sees PYTHONHOME, but it is unset afterwards . In bash:

```
[[ -n $CONDA_PREFIX ]] || CONDA_PREFIX=/Users/mypath/anaconda
alias idl='export PYTHONHOME=$CONDA_PREFIX ; $IDL_DIR/bin/idl ; unset PYTHONHOME'
```

this is the only thing that really worked for me, without breaking stuff left and right.

On Wednesday, 11 January 2017 23:34:41 UTC+1, ivol...@gmail.com wrote:

> 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-installing-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.

Subject: Re: IDL Python Bridge
Posted by joe.llama on Thu, 02 Feb 2017 21:14:29 GMT
View Forum Message <> Reply to Message

Can anyone help me out with getting the idl-python bridge to work? I have followed the instructions in a post above and I can call python from IDL but I can't call idl from within python .

In Anaconda Python 2.7 if I get:

In [1]: from idlpy import *
Fatal Python error: PyThreadState_Get: no current thread
Abort trap: 6

I'm running idl 8.5.1 on OS X 10.12 with Anaconda Python 2.7.

Thanks!

Subject: Re: IDL Python Bridge
Posted by cwamsley on Mon, 28 Aug 2017 13:10:38 GMT

View Forum Message <> Reply to Message

On Thursday, February 2, 2017 at 4:14:30 PM UTC-5, joe....@lowell.edu wrote:
> Can anyone help me out with getting the idl-python bridge to work? I have followed the instructions in a post above and I can call python from IDL but I can't call idl from within python .
>
> In Anaconda Python 2.7 if I get:
>
> In [1]: from idlpy import *
> Fatal Python error: PyThreadState_Get: no current thread
> Abort trap: 6
>
> I'm running idl 8.5.1 on OS X 10.12 with Anaconda Python 2.7.
>
> Thanks!

I received the same error working on Win 7, Anaconda 2 and Python 2.7. IDL was also crashing when I tried to start the IDL/Python bridge. Both problems were corrected by removing unnecessary spaces I had in the PYTHONPATH environment variable (e.g. bad: C:\Anaconda2; C:\Anaconda2\Tools\Scripts | good C:\Anaconda2;C:\Anaconda2\Tools\Scripts). Also make sure to run the python setup.py install command from the IDL bridges directory as indicated in the Harris help information.

Chuck