

Hi All,

Tonight I thought I'd take a look at the IDL-Python bridge - it looks pretty interesting and it could potentially unlock a lot of functionality from the python world for idl users.

I started by trying to get IDL 8.5 to talk to Python 2.7 (default Mac OS X installation), and failed. Then searched around and found some posts by Chris Torrence on this newsgroup and on stack overflow. Following advice from these threads I installed Anaconda, learned something about setting up different python environments (and installed python 3.4) and battled on, modifying .bashrc, setting environment variables, running otool, trying to follow the IDL help instructions... but ultimately failing to get it to work.

After a while I thought I'd try in IDL 8.6.1 - supposedly improved with respect to setting up the bridge, and compatible with python 3.6. Once more I followed the instructions in the IDL8.6 help, seemed to get close at times but it's still not working correctly. I can call IDL from python and execute the example code from IDL help as follows:

```
>>> from idlpy import *
>>> import numpy.random as ran
>>> arr = ran.rand(100)
>>> p = IDL.plot(arr, title='My Plot')
>>> p.color = 'red'
>>> p.save('myplot.pdf')
>>> p.close()
```

but what I would prefer is to be able to call python from IDL. Executing the code from IDL Help gives the following error:

```
IDL> ran = Python.Import('numpy.random')
% DLM_LOAD: Error loading sharable executable.
    Symbol: IDL_Load, File = /usr/local/harris/idl86/bin/bin.darwin.x
    86_64/idl_python36.so
    dlopen(/usr/local/harris/idl86/bin/bin.darwin.x86_64/idl_python36
    .so, 1): Library not loaded: @rpath/libpython3.6m.dylib
    Referenced from: /usr/local/harris/idl86/bin/bin.darwin.x86_64/
    idl_python36.so
    Reason: image not found
% Execution halted at: $MAIN$
```

If anyone can help decipher this I'd appreciate it. At this stage I'm not all that impressed with the process of setting this up.

Also, I've noticed that if I'm connected to the internet IDL8.6+ takes a \*long\* time to start up. If I disconnect from the internet it starts up straight away. Looks like the new IDL is calling home to

check those licenses...

thanks in advance,

Will.

---

Subject: Re: python bridge setup - IDL8.6.1 Mac OS X Sierra  
Posted by [chris\\_torrence@NOSPAM](#) on Fri, 18 Aug 2017 20:31:48 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hi Will,

It looks like something changed with Anaconda between 3.5 and 3.6, where it now has the @rpath in the library name. When you run the IDL\_DIR/lib/bridges/setup.py command, it is supposed to modify your IDL python binaries with the Anaconda location. But it fails because it isn't expecting that "@rpath".

So, I've modified the setup.py to now handle both cases. If you want to try this, open up IDL\_DIR/lib/bridges/setup.py and modify lines 92-102:

```
try:
    cmd = ["install_name_tool", "-change", pylib, pylibpath, pythonidllibpath]
    print("\n" + " ".join(cmd))
    check_call(cmd)
    cmd = ["install_name_tool", "-change", "@rpath/" + pylib, pylibpath, pythonidllibpath]
    print("\n" + " ".join(cmd))
    check_call(cmd)
    cmd = ["install_name_tool", "-change", pylib, pylibpath, idlpythonlibpath]
    print("\n" + " ".join(cmd))
    check_call(cmd)
    cmd = ["install_name_tool", "-change", "@rpath/" + pylib, pylibpath, idlpythonlibpath]
    print("\n" + " ".join(cmd))
    check_call(cmd)
except Exception as e:
    print("\n*** install_name_tool failed.")
    print("\n*** Check the paths above, or run the script as 'sudo python setup.py'.")
    exit()
```

Then run "python setup.py install". Hopefully you should then be able to call Python from within IDL.

Let me know if this works.

Cheers,  
Chris

---

Subject: Re: python bridge setup - IDL8.6.1 Mac OS X Sierra

Posted by [wallabadah](#) on Sat, 19 Aug 2017 10:11:55 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Thanks for the quick response Chris, it's now much closer to working as it should, but not quite...

Running IDL code from python works as expected, using the following code from IDL Help (albeit with an error

```
>>> from idlpy import *
>>> import numpy.random as ran
>>> arr = ran.rand(100)
>>> p = IDL.plot(arr, title='My Plot')
(a plot is displayed)
Warning: Cannot convert string "-adobe-helvetica-medium-r-normal-*-*-120-75-75-p-*-iso8859-1"
to type FontStruct
```

Running python within IDL works as expected, using the following code form IDL help:

```
>>>
import matplotlib.pyplot as plt
import numpy.random as ran
arr = ran.rand(100)
p = plt.plot(arr)
plt.show()
(a plot is displayed)
```

But running python code at the IDL command line doesn't work:

```
ran = Python.Import('numpy.random')
arr = ran.rand(100) ; call "rand" method
plt = Python.Import('matplotlib.pyplot')
p = plt.plot(arr) ; call "plot", pass an array
void = plt.show(block=0) ; pass keyword
(no plot is shown, no error messages)
```

If you can provide any advice Chris I'd appreciated it, and I'm sure others on the list would appreciate it too.

On a related note, is there any documentation on how using one of the IDL bridges works with distributing software? If one was to develop an IDL application that used (for example) some python code via a bridge, and tried to distribute it using make\_rt and the IDL virtual machine... I guess the end user would have to go through the process of setting paths and installing python just as I have been doing in the last few days? Seems a little tricky for the non-specialist... Are there plans to streamline the process?

thanks,

Will

---

---

Subject: Re: python bridge setup - IDL8.6.1 Mac OS X Sierra  
Posted by [chris\\_torrence@NOSPAM](#) on Fri, 25 Aug 2017 14:30:12 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Saturday, August 19, 2017 at 4:11:59 AM UTC-6, wallabadah wrote:

> Thanks for the quick response Chris, it's now much closer to working as it should, but not quite...

>  
> Running IDL code from python works as expected, using the following code from IDL Help (albeit with an error

>  
>>>> from idlpy import \*  
>>>> import numpy.random as ran  
>>>> arr = ran.rand(100)  
>>>> p = IDL.plot(arr, title='My Plot')  
> (a plot is displayed)  
> Warning: Cannot convert string "  
-adobe-helvetica-medium-r-normal-\*-\*120-75-75-p-\*iso8859-1 " to type FontStruct

>  
> Running python within IDL works as expected, using the following code form IDL help:

>  
>>>>  
> import matplotlib.pyplot as plt  
> import numpy.random as ran  
> arr = ran.rand(100)  
> p = plt.plot(arr)  
> plt.show()  
> (a plot is displayed)

>  
> But running python code at the IDL command line doesn't work:

>  
> ran = Python.Import('numpy.random')  
> arr = ran.rand(100) ; call "rand" method  
> plt = Python.Import('matplotlib.pyplot')  
> p = plt.plot(arr) ; call "plot", pass an array  
> void = plt.show(block=0) ; pass keyword  
> (no plot is shown, no error messages)

>  
> If you can provide any advice Chris I'd appreciate it, and I'm sure others on the list would appreciate it too.

>  
> On a related note, is there any documentation on how using one of the IDL bridges works with distributing software? If one was to develop an IDL application that used (for example) some python code via a bridge, and tried to distribute it using make\_rt and the IDL virtual machine... I guess the end user would have to go through the process of setting paths and installing python just as I have been doing in the last few days? Seems a little tricky for the non-specialist... Are there plans to streamline the process?

>  
> thanks,

>  
> Will

Hi Will,

Is the IDL->Python bridge working for non-graphics? In other words, if you just do:

```
IDL> >>>2+2
```

Does it print out the answer?

If so, I'd suggest it's just a problem with either the Qt within Anaconda, or running Qt from within Python within IDL. I can't get the plot to show up on my Windows box either (I get a weird Qt error), although everything else works fine.

Regarding your "distributing software" question, we don't have any documentation or pre-canned solution for making this work. That's one of the unfortunate side effects of working with an open-source 3rd party package like Python - you sort of get what you pay for. Anyway, it's a great suggestion and we can think about adding better support in the future.

Cheers,  
Chris

---

Subject: Re: python bridge setup - IDL8.6.1 Mac OS X Sierra

Posted by [wallabadah](#) on Tue, 29 Aug 2017 11:01:17 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Thanks for the response Chris, much appreciated.

Yes, the IDL->Python bridge seems to work for non-graphics - the following code works as expected:

```
IDL> >>>2+2
```

```
% Loaded DLM: PYTHON36.
```

```
4
```

```
IDL>
```

Having a solution for distribution would be great, although I realise it's probably a lot more difficult than it sounds. At this stage incorporating Python code into commercial software would not really be an option for us - I feel like it would be too much to ask end users who are non-specialists to go through the python bridge setup and configuration. MAKE\_RT has made distribution of IDL software so much easier, but this would be a step backward.

thanks for your help,

Will.

---