Subject: Re: IDL 8.5 Python Bridge
Posted by Jim Pendleton on Sun, 23 Aug 2015 21:58:40 GMT
View Forum Message <> Reply to Message

On Sunday, August 23, 2015 at 1:49:53 PM UTC-6, robintw wrote:
> Hi all (and particularly Chris),
>
> Is there a way to deal with output parameters from IDL functions/procedures through the Python bridge?
>
> For example, I am calling the ENVI_ENTER_DATA procedure (see docs at http://www.exelisvis.com/docs/ENVI_ENTER_DATA.html) which works fine, but I'd like to get the value of the output parameter r_fid.
>
> I've tried:
>
> IDL.ENVI_ENTER_DATA(arr, r_fid=result)
>
> and get a message saying that result isn't defined (obviously). I've then tried initialising result (eg. to zero) first, but the value doesn't seem to be changed.
>
> Any ideas? Is this possible?
>
> Cheers,
>
> Robin
>
> On Tuesday, 18 August 2015 22:42:19 UTC+1, Jim P wrote:
>> On Thursday, August 13, 2015 at 3:39:33 PM UTC-6, Chris Torrence wrote:
>>> On Thursday, August 13, 2015 at 12:33:33 PM UTC-6, chu xiangning wrote:
>>>> On Thursday, August 6, 2015 at 9:46:15 AM UTC-7, Chris Torrence wrote:
>>>> > Hi all,
>>>> >
>>>> > As mentioned in my previous post, IDL 8.5 (now available unofficially) contains the new IDL Python bridge. I'll attach the "What's New" to the end of this post.
>>>> >
>>>> > The actual documentation for the Python bridge will be available on the website in a few weeks. It has a few updates (especially for configuration and the IPython notebook) that didn't make it onto the DVD. If you would like a copy now, please email me directly, chris <dot> torrence <at> harris <dot> com
>>>> >
>>>> > Also, we are looking for feedback on the Python configuration. Because both IDL and Python are large, we did not try to combine the two. You will need to install Python and Numpy (preferably from Anaconda) and then set several environment variables to get them to play nicely. Mac is especially tricky because Python has a lot of conflicts with the Macintosh system libraries. We'd like feedback on what worked (or didn't work) and how we can make it more streamlined in the future.
>>>> >

>>>> > Thanks!
>>>> >
>>>> > -Chris
>>>> >
>>>> >
>>>> > Here's the "What's New" page for the bridge:
>>>> >
>>>> > From your IDL code, you can now access any Python modules, transfer variables, and call built-in functions. Similarly, from your Python code, you can make IDL calls, transfer variables, and manipulate IDL objects. The bridge has the following features:
>>>> > * Works with Python 2.7+ and Python 3.4+
>>>> > * Access to all IDL routines and Python modules
>>>> > * Seamless: looks just like an IDL object or Python module
>>>> > * All bridge output is redirected to the standard output
>>>> > * Case sensitivity and row/column major is handled automatically
>>>> > * Can execute arbitrary command strings in either language
>>>> > * Automatic data conversion from IDL arrays to numpy arrays
>>>> > * Data is passed by reference when calling routines/methods
>>>> > * Can pass main variables back & forth
>>>> > * IDL IPython Notebook Kernel
>>>> >
>>>> > For example, within IDL, you could execute the following Python commands to create a matplotlib plot:
>>>> > IDL> ran = Python.Import('numpy.random')
>>>> > IDL> arr = ran.rand(100) ; call "rand" method
>>>> > IDL> plt = Python.Import('matplotlib.pyplot')
>>>> > IDL> p = plt.plot(arr)   ; call "plot", pass an array
>>>> > IDL> void = plt.show(block=0) ; pass keyword
>>>> >
>>>> > Within IDL, you can also directly enter Python "command-line mode":
>>>> > IDL> >>>
>>>> > >>> import matplotlib.pyplot as plt
>>>> > >>> import numpy.random as ran
>>>> > >>> arr = ran.rand(100)
>>>> > >>> p = plt.plot(arr)
>>>> > >>> plt.show()
>>>> > >>>
>>>> > IDL>
>>>> >
>>>> > On the Python side, you can easily access all IDL functionality:
>>>> > >>> from idlpy import IDL
>>>> > >>> 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()
>>>>

>>>> Great news!
>>>> With the bridge I will be able to use IDL within ipython notebook!
>>>
>>> Yes, we now have an IPython Notebook kernel for IDL. The documentation will be up on the website soon, but in the meantime, you can email me for a PDF, at chris <dot> torrence <at> harris <dot> com
>>>
>>> -Chris
>>
>> Looks like the new IDL 8.5 docs are now published online.
>>
>> http://www.exelisvis.com/docs/using_idl_home.html

To return to Python those IDL keywords that are populated on output, you may want to use a multi-step process. For example,

```
IDL.arr = arr
IDL.run("ENVI_ENTER_DATA, arr, r_fid=result")
r = IDL.result
```

Jim P.