Subject: IDL 8.5 Python Bridge
Posted by chris_torrence@NOSPAM on Thu, 06 Aug 2015 16:46:12 GMT
View Forum Message <> Reply to Message

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()

Subject: Re: IDL 8.5 Python Bridge
Posted by Jim Pendleton on Thu, 06 Aug 2015 18:09:00 GMT
View Forum Message <> Reply to Message

On Thursday, August 6, 2015 at 10:46:15 AM UTC-6, 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()

And by "now available unofficially" you may ask your sales rep for information how you may obtain it if your IDL and/or ENVI support under current maintenance, and if you can't find the necessary info on your exelisvis.com online account.

Keep Chris busy with your feedback!

Jim P.

Subject: Re: IDL 8.5 Python Bridge
Posted by pfp on Thu, 06 Aug 2015 20:35:00 GMT
View Forum Message <> Reply to Message

Thanks, Chris this seems very nicely done! The seamless use of one language's objects in the other one will make my life so much easier.

Are there any estimates on the release of 8.5?

Paulo

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()

Subject: Re: IDL 8.5 Python Bridge
Posted by chris_torrence@NOSPAM on Thu, 06 Aug 2015 21:30:21 GMT
View Forum Message <> Reply to Message

On Thursday, August 6, 2015 at 2:35:03 PM UTC-6, pfpen...@gmail.com wrote:
> Thanks, Chris this seems very nicely done! The seamless use of one language's objects in the other one will make my life so much easier.
>
> Are there any estimates on the release of 8.5?
>
> Paulo
>
> 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()

Hi Paulo,
IDL 8.5 and ENVI 5.3 are up on the website as we speak. The documentation isn't up yet.
Cheers,
Chris

Subject: Re: IDL 8.5 Python Bridge
Posted by pfp on Thu, 06 Aug 2015 22:02:31 GMT
View Forum Message <> Reply to Message

Thanks! I had missed the other post about its release status. Downloading it now...

On Thursday, August 6, 2015 at 2:31:29 PM UTC-7, Chris Torrence wrote:
> On Thursday, August 6, 2015 at 2:35:03 PM UTC-6, pfpen...@gmail.com wrote:
>> Thanks, Chris this seems very nicely done! The seamless use of one language's objects in the other one will make my life so much easier.
>>
>> Are there any estimates on the release of 8.5?
>>
>> Paulo
>>
>> 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()
>
> Hi Paulo,
> IDL 8.5 and ENVI 5.3 are up on the website as we speak. The documentation isn't up yet.
> Cheers,
> Chris

Subject: Re: IDL 8.5 Python Bridge
Posted by robintw on Tue, 11 Aug 2015 10:21:22 GMT
View Forum Message <> Reply to Message

Thanks - this looks great! I do a lot of work in both Python and IDL, and this will help a lot.

Out of interest, will it be possible to use ENVI API from within Python using this bridge? And if so, would both the new and old APIs work? As they're just IDL functions and objects, I'd assume they would work - but I wondered if there would be issues if I tried to use the GUI manipulation functions (eg. to view images).

Cheers,

Robin

On Thursday, 6 August 2015 17:46:15 UTC+1, 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()

Subject: Re: IDL 8.5 Python Bridge
Posted by chris_torrence@NOSPAM on Tue, 11 Aug 2015 23:10:00 GMT
View Forum Message <> Reply to Message

On Tuesday, August 11, 2015 at 4:21:24 AM UTC-6, robintw wrote:
> Thanks - this looks great! I do a lot of work in both Python and IDL, and this will help a lot.
>
> Out of interest, will it be possible to use ENVI API from within Python using this bridge? And if so, would both the new and old APIs work? As they're just IDL functions and objects, I'd assume they would work - but I wondered if there would be issues if I tried to use the GUI manipulation functions (eg. to view images).
>
> Cheers,
>
> Robin
>

Hi Robin,

In general, you can use any IDL or ENVI routines from within Python. However, sometimes it seems like IDL/ENVI widget applications don't play nicely with Python and Python will crash. I haven't done too much digging to find out exactly why, but if you encounter any issues, definitely post them on the newsgroup.

Cheers,
Chris

Subject: Re: IDL 8.5 Python Bridge
Posted by robintw on Wed, 12 Aug 2015 07:14:39 GMT
View Forum Message <> Reply to Message

Thanks Chris - I'll post if I find any problems.

(Now I just need to persuade my university's IT services department to download a copy for me...because of course they have the account with Excelis)

Robin

On Wednesday, 12 August 2015 00:10:04 UTC+1, Chris Torrence wrote:
> On Tuesday, August 11, 2015 at 4:21:24 AM UTC-6, robintw wrote:
>> Thanks - this looks great! I do a lot of work in both Python and IDL, and this will help a lot.
>>
>> Out of interest, will it be possible to use ENVI API from within Python using this bridge? And if so, would both the new and old APIs work? As they're just IDL functions and objects, I'd assume they would work - but I wondered if there would be issues if I tried to use the GUI manipulation functions (eg. to view images).
>>

>> Cheers,
>>
>> Robin
>>
>
> Hi Robin,
>
> In general, you can use any IDL or ENVI routines from within Python. However, sometimes it seems like IDL/ENVI widget applications don't play nicely with Python and Python will crash. I haven't done too much digging to find out exactly why, but if you encounter any issues, definitely post them on the newsgroup.
>
> Cheers,
> Chris

Subject: Re: IDL 8.5 Python Bridge
Posted by PMallas on Thu, 13 Aug 2015 18:16:14 GMT
View Forum Message <> Reply to Message

On Thursday, August 6, 2015 at 12:46:15 PM UTC-4, 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()

This sounds great - I am very excited to check this out. Like many others here, I have been using python along with IDL.

Since you have decided to bridge these two languages, would it be possible to bridge the development environment too? Say, by integrating PyDev with your eclipse IDE? Seems like an easy thing to do and relieves someone like me from having to maintain an additional development environment just for python. Just a thought ....

Subject: Re: IDL 8.5 Python Bridge
Posted by natha on Thu, 13 Aug 2015 18:31:24 GMT
View Forum Message <> Reply to Message

Hi Chris,

I was wondering if a python drawable wrapper object will also be available. Like a pyhton object to which IDL could draw.

Is this asking too much?
nata

Subject: Re: IDL 8.5 Python Bridge
Posted by chuxiangning on Thu, 13 Aug 2015 18:33:29 GMT
View Forum Message <> Reply to Message

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!

Subject: Re: IDL 8.5 Python Bridge
Posted by chris_torrence@NOSPAM on Thu, 13 Aug 2015 21:37:57 GMT
View Forum Message <> Reply to Message

On Thursday, August 13, 2015 at 12:31:28 PM UTC-6, nata wrote:
> Hi Chris,
>
> I was wondering if a python drawable wrapper object will also be available. Like a pyhton object to which IDL could draw.

>
> Is this asking too much?
> nata

Hi nata,

You mean like this?

```
; Define some IDL variables
labels = ['Baltam', 'Python', 'IDL', 'Other']
sizes = [20, 30, 40, 10]
colors = ['yellowgreen', 'gold', 'lightskyblue', 'lightcoral']
explode = [0, 0, 0.1, 0] ; "explode" the 3rd slice
; Import some Python modules
pyplot = Python.Import('matplotlib.pyplot')
; Call methods on the Python modules
pie = pyplot.pie(sizes, explode=explode, $
labels=labels, colors=colors, $
autopct='%1.1f%%', /shadow, startangle=90)
void = pyplot.axis('equal')
void = pyplot.savefig("myplot.png", dpi = 96)
void = pyplot.show()
```

If so, then the answer is yes! Forget those pesky IDL function graphics, now you can use Python function graphics! ;-)

-Chris

Subject: Re: IDL 8.5 Python Bridge
Posted by chris_torrence@NOSPAM on Thu, 13 Aug 2015 21:39:32 GMT
View Forum Message <> Reply to Message

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

Subject: Re: IDL 8.5 Python Bridge
Posted by penteado on Fri, 14 Aug 2015 02:40:48 GMT
View Forum Message <> Reply to Message

On Thursday, August 13, 2015 at 2:37:59 PM UTC-7, Chris Torrence wrote:
> If so, then the answer is yes! Forget those pesky IDL function graphics, now you can use Python function graphics! ;-)

One of the main uses I have for the bridge is to make IDL graphics (Function Graphics, iTools, Coyote Graphics) from Python, so that I do not have to deal with matplotlib...

Subject: Re: IDL 8.5 Python Bridge
Posted by Jim Pendleton on Tue, 18 Aug 2015 21:42:17 GMT
View Forum Message <> Reply to Message

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

Subject: Re: IDL 8.5 Python Bridge
Posted by robintw on Sun, 23 Aug 2015 19:49:50 GMT
View Forum Message <> Reply to Message

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

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.