Subject: Slither with matplotlib.mlab.PCA makes IDL crash
Posted by Fabzi on Wed, 25 Jun 2014 14:18:02 GMT
View Forum Message <> Reply to Message

Hi all,

I am using the Slither dlm (python3.3, linux_x64). I know this is not precisely an IDL question, but does anyone have an idea how to solve the following problem?

The matplotlib.mlab.PCA function (http://matplotlib.org/api/mlab_api.html#) works fine under my python distrib.

I wanted to call it from IDL but this makes IDL chrash quite badly.

After several tests I noticed that it is not in the linkage between IDL and python but rather in python (called from IDL) that the problem comes from. To come to this conclusion, I made a small script.

The following works fine in python3:

```
import matplotlib.mlab as mlab
import numpy as np
import random as r
n = 50
a = np.array([r.random() for n in range(n)]).reshape(1,n)
b = np.array([r.random() for n in range(n)]).reshape(1,n)
r = np.concatenate((a,b), 0).transpose()
p = mlab.PCA(r)
[/python code]
```

But the exact same called called from with IDL with the exec() syntax crashes:

```
py = pyimport('__main__')

py->exec, 'print(sys.version)'
; LOOKS OK: 3.3.2+ (default, Feb 28 2014, 00:56:04)

py->exec, 'import matplotlib.mlab as mlab'
py->exec, 'import numpy as np'
py->exec, 'import random as r'
py->exec, 'n = 50'
py->exec, 'a = np.array([r.random() for n in range(n)]).reshape(1,n)'
py->exec, 'b = np.array([r.random() for n in range(n)]).reshape(1,n)'
py->exec, 'r = np.concatenate((a,b), 0).transpose()'
```

py->exec, 'p = mlab.PCA(r)' ; it crashes here

Any hint on where I should look at first?

Thanks!

Fabien

Subject: Re: Slither with matplotlib.mlab.PCA makes IDL crash
Posted by Fabzi on Fri, 27 Jun 2014 13:17:45 GMT
View Forum Message <> Reply to Message

Folks,

I tried the code on python 2.7 as well without success. If someone has
slither, numpy and matplotlib installed I would be really interested to
know what happens when the following code is executed on another machine
(works for python 2.7 and 3.3):

```
IDL> py = pyimport('__main__')
py->exec, 'import matplotlib.mlab as mlab'
py->exec, 'import numpy as np'
py->exec, 'import random as r'
py->exec, 'n = 50'
py->exec, 'a = np.array([r.random() for i in range(n)]).reshape(1,n)'
py->exec, 'b = np.array([r.random() for i in range(n)]).reshape(1,n)'
py->exec, 'r = np.concatenate((a,b), 0).transpose()'
py->exec, 'p = mlab.PCA(r)' ; it crashes here
```

thanks a lot for your help!

Fabien

On 25.06.2014 16:18, Fabien wrote:
> Hi all,
>
> I am using the Slither dlm (python3.3, linux_x64). I know this is not
> precisely an IDL question, but does anyone have an idea how to solve the
> following problem?
>
> The matplotlib.mlab.PCA function
> (http://matplotlib.org/api/mlab_api.html#) works fine under my python
> distrib.
>
> I wanted to call it from IDL but this makes IDL chrash quite badly.
>

> After several tests I noticed that it is not in the linkage between IDL
> and python but rather in python (called from IDL) that the problem comes
> from. To come to this conclusion, I made a small script.
>
> The following works fine in python3:
>
>
> import matplotlib.mlab as mlab
> import numpy as np
> import random as r
> n = 50
> a = np.array([r.random() for n in range(n)]).reshape(1,n)
> b = np.array([r.random() for n in range(n)]).reshape(1,n)
> r = np.concatenate((a,b), 0).transpose()
> p = mlab.PCA(r)
> [/python code]
>
> But the exact same called called from with IDL with the exec() syntax
> crashes:
>
> py = pyimport('__main__')
>
> py->exec, 'print(sys.version)'
> ; LOOKS OK: 3.3.2+ (default, Feb 28 2014, 00:56:04)
>
> py->exec, 'import matplotlib.mlab as mlab'
> py->exec, 'import numpy as np'
> py->exec, 'import random as r'
> py->exec, 'n = 50'
> py->exec, 'a = np.array([r.random() for n in range(n)]).reshape(1,n)'
> py->exec, 'b = np.array([r.random() for n in range(n)]).reshape(1,n)'
> py->exec, 'r = np.concatenate((a,b), 0).transpose()'
> py->exec, 'p = mlab.PCA(r)' ; it crashes here
>
> Any hint on where I should look at first?
>
> Thanks!
>
> Fabien

Subject: Re: Slither with matplotlib.mlab.PCA makes IDL crash
Posted by Phillip Bitzer on Fri, 27 Jun 2014 15:15:17 GMT
View Forum Message <> Reply to Message

Well, I don't know if this is a good or bad thing, but I'm able to run it successfully. No crash.

IDL> py->exec, 'print p.mu'

```
[ 0.50734012  0.50217405]

IDL> py -> exec, "import sys"
IDL> py->exec, 'print(sys.version)'
2.7.5 (default, Mar  9 2014, 22:15:05)
[GCC 4.2.1 Compatible Apple LLVM 5.0 (clang-500.0.68)]

IDL> help, !VERSION
** Structure !VERSION, 8 tags, length=104, data length=100:
   ARCH            STRING    'x86_64'
   OS              STRING    'darwin'
   OS_FAMILY       STRING    'unix'
   OS_NAME         STRING    'Mac OS X'
   RELEASE         STRING    '8.2.3'
   BUILD_DATE      STRING    'May  2 2013'
   MEMORY_BITS     INT             64
   FILE_OFFSET_BITS
                   INT             64
```

Subject: Re: Slither with matplotlib.mlab.PCA makes IDL crash
Posted by Fabzi on Fri, 27 Jun 2014 16:31:51 GMT

View Forum Message <> Reply to Message

Thanks Philipp for your feedback. I get a segmentation fault on linux. I think I'll just have to use a workaround with external files, it should be ok.

Have a nice week-end all!

On 27.06.2014 17:15, Phillip Bitzer wrote:

```
>  Well, I don't know if this is a good or bad thing, but I'm able to run it successfully. No crash.
>
>  IDL> py->exec, 'print p.mu'
>  [ 0.50734012  0.50217405]
>
>  IDL> py -> exec, "import sys"
>  IDL> py->exec, 'print(sys.version)'
>  2.7.5 (default, Mar  9 2014, 22:15:05)
>  [GCC 4.2.1 Compatible Apple LLVM 5.0 (clang-500.0.68)]
>
>  IDL> help, !VERSION
>  ** Structure !VERSION, 8 tags, length=104, data length=100:
>     ARCH            STRING    'x86_64'
>     OS              STRING    'darwin'
>     OS_FAMILY       STRING    'unix'
>     OS_NAME         STRING    'Mac OS X'
>     RELEASE         STRING    '8.2.3'
```

> BUILD_DATE STRING 'May 2 2013'
> MEMORY_BITS INT 64
> FILE_OFFSET_BITS
> INT 64
>

Subject: Re: Slither with matplotlib.mlab.PCA makes IDL crash
Posted by JP on Fri, 03 Oct 2014 00:19:40 GMT
View Forum Message <> Reply to Message

Thanks Chris,

I'm using IDL in a supercomputer and it has 8.2 :((by the way, it's listed as a "graphics package" wtf?) http://nci.org.au/nci-systems/national-facility/peak-system/ raijin/application-software/
Will check what are the chances of upgrading to 8.3 but I doubt it will be possible in the short term.
One of the reasons I need to use this python bridge is to be able to read .vrt files (see https://groups.google.com/d/msg/comp.lang.idl-pvwave/YF2EaUq nP_k/bhMcY_Iqk8MJ) but now I'm stuck again :(

JP

On Friday, 3 October 2014 01:33:39 UTC+10, Chris Torrence wrote:
> On Thursday, October 2, 2014 5:24:36 AM UTC-6, JP wrote:
>
>> well,
>
>>
>
>> my IDL didn't crash with your code, Have you tried asking the people in www.jacquette.com ?
>
>>
>
>>
>
>>
>
>> JP
>
>>
>
>>
>
>>
>

>>
>
>>
>
>>
>
>>
>
>> [jpg599@raijin2 ~]$ idl
>
>>
>
>> IDL Version 8.2.1 (linux x86_64 m64). (c) 2012, Exelis Visual Information Solutions, Inc.
>
>>
>
>> Installation number: 237570.
>
>>
>
>> Licensed for use by: ANU Supercomputer Facility
>
>>
>
>>
>
>>
>
>> IDL> py = pyimport('__main__')
>
>>
>
>> py->exec, 'import matplotlib.mlab as mlab'
>
>>
>
>> py->exec, 'import numpy as np'
>
>>
>
>> py->exec, 'import random as r'
>
>>
>
>> py->exec, 'n = 50'
>
>>
>

>> py->exec, 'a = np.array([r.random() for i in range(n)]).reshape(1,n)'
>
>>
>
>> py->exec, 'b = np.array([r.random() for i in range(n)]).reshape(1,n)'
>
>>
>
>> py->exec, 'r = np.concatenate((a,b), 0).transpose()'
>
>>
>
>> Slither 3.0, by Jacquette Consulting, Inc. Built on 2014-02-03 11:40:48.314070
>
>>
>
>> Copyright (c) 2011 by Jacquette Consulting, Inc. All rights reserved.
>
>>
>
>> http://www.jacquette.com
>
>>
>
>>
>
>>
>
>> Jacquette Consulting, Inc. grants you a non-exclusive and non-transferable
>
>>
>
>> license to use Slither (hereinafter "the software").
>
>>
>
>>
>
>>
>
>> The granting of this license does not obligate Jacquette Consulting, Inc. to
>
>>
>
>> provide any sort of technical support.
>
>>
>

>>
>
>>
>
>> THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
OR
>
>>
>
>> IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
>
>>
>
>> FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT
SHALL THE
>
>>
>
>> AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR
OTHER
>
>>
>
>> LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM,
>
>>
>
>> OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN THE
>
>>
>
>> SOFTWARE.
>
>>
>
>> % Loaded DLM: SLITHER.
>
>>
>
>> IDL> py->exec, 'import matplotlib.mlab as mlab'
>
>>
>
>> IDL> py->exec, 'import numpy as np'
>
>>

>
>> IDL> py->exec, 'import random as r'
>
>>
>
>> IDL> py->exec, 'n = 50'
>
>>
>
>> IDL> py->exec, 'a = np.array([r.random() for i in range(n)]).reshape(1,n)'
>
>>
>
>> IDL> py->exec, 'b = np.array([r.random() for i in range(n)]).reshape(1,n)'
>
>>
>
>> IDL> py->exec, 'r = np.concatenate((a,b), 0).transpose()'
>
>>
>
>> IDL> py->exec, 'p = mlab.PCA(r)' ; it crashes here
>
>>
>
>> IDL>
>
>>
>
>>
>
>>
>
>>
>
>>
>
>>
>
>>
>
>> On Thursday, 2 October 2014 20:04:45 UTC+10, Fabien wrote:
>
>>
>
>>> Hi JP,
>
>>

>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> On 02.10.2014 04:01, JP wrote:
>
>>
>
>>>
>
>>
>
>>>> So i get an error when trying to do 'a = np.array([r.random() for n in range(n)]).reshape(1,n)'
>
>>
>
>>>
>
>>
>
>>>> but in any case, no crash when I get to the mlab.PCA.
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> yes this was a stupid error in the python code. This should work better:
>
>>

```
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>  py = pyimport('__main__')
>
>>
>
>>>
>
>>
>
>>>  py->exec, 'import matplotlib.mlab as mlab'
>
>>
>
>>>
>
>>
>
>>>  py->exec, 'import numpy as np'
>
>>
>
>>>
>
>>
>
>>>  py->exec, 'import random as r'
>
>>
>
>>>
>
>>
>
>>>  py->exec, 'n = 50'
>
>>
```

>
>>>
>
>>
>
>>> py->exec, 'a = np.array([r.random() for i in range(n)]).reshape(1,n)'
>
>>
>
>>>
>
>>
>
>>> py->exec, 'b = np.array([r.random() for i in range(n)]).reshape(1,n)'
>
>>
>
>>>
>
>>
>
>>> py->exec, 'r = np.concatenate((a,b), 0).transpose()'
>
>>
>
>>>
>
>>
>
>>> py->exec, 'p = mlab.PCA(r)' ; it crashes here
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> And this still makes IDL crash on my machine. I gave up on using slither
>
>>

>
>>>
>
>>
>
>>> long ago now, which is a pity since I'd like to play around with python
>
>>
>
>>>
>
>>
>
>>> in IDL much more often...
>
>
>
> Hi Fabien and JP,
>
>
>
> I've played around with Slither a bit, and I have some suggestions. First, I would strongly recommend not bothering to try it in IDL 8.2.x or earlier. It was compiled against IDL 8.3, and if you can somehow get it to work with an earlier version, you are getting extremely lucky. If it crashes against IDL 8.2.x, I wouldn't even try to dig deeper. This is because the symbol tables tend to shift around with each new release of IDL, and your DLM ends up pointing at the wrong symbols (IDL_Deltmp instead of IDL_Gettmp, for example :-).
>
>
>
> With IDL 8.4, we tried very hard to keep the symbol table the same as IDL 8.3, specifically so that Slither would keep working - so you should be able to upgrade to IDL 8.4 (in a few weeks) without any problems.
>
>
>
> Second, I think using "Exec" is not the right approach. Slither was really designed to mimic the Python API from within IDL. For example, I would rewrite your code as:
>
>
>
> mlab = pyimport("matplotlib.mlab")
>
> np = pyimport("numpy")
>
> r = pyimport("numpy.random")
>
> n = 50

>
> a = r.rand(n) ; why not use IDL's mersenne twister??
>
> b = r.rand(n)
>
> c = TRANSPOSE([[a],[b]])
>
> p = mlab.PCA(c)
>
>
>
> Now, obviously if you want to use things like Python generators or Lambda functions, you will need to use Exec. But in general, it will make your code cleaner and easier to debug if you simply treat the Python objects just like IDL objects and call methods directly on the imported libraries.
>
>
>
> Hope this helps a bit.
>
>
>
> Cheers,
>
> Chris
>
> ExelisVIS

Subject: Re: Slither with matplotlib.mlab.PCA makes IDL crash
Posted by chris_torrence@NOSPAM on Fri, 03 Oct 2014 01:50:29 GMT
View Forum Message <> Reply to Message

On Thursday, October 2, 2014 6:19:44 PM UTC-6, JP wrote:
> Thanks Chris,
>
>
>
> I'm using IDL in a supercomputer and it has 8.2 :((by the way, it's listed as a "graphics package" wtf?) http://nci.org.au/nci-systems/national-facility/peak-system/raijin/application-software/
>
> Will check what are the chances of upgrading to 8.3 but I doubt it will be possible in the short term.
>
> One of the reasons I need to use this python bridge is to be able to read .vrt files (see https://groups.google.com/d/msg/comp.lang.idl-pvwave/YF2EaUq nP_k/bhMcY_Iqk8MJ) but now I'm stuck again :(
>

>
>
> JP
>
>

Ah, I see. Bummer. You might want to contact Jaquette Consulting and see if they still have a version of Slither built against IDL 8.2. I know that they had that at some point, and maybe they would be willing to just give you the older version.

Good luck!
-Chris
