Subject: Re: iimage tool
Posted by David Fanning on Tue, 15 Jun 2004 14:35:32 GMT
View Forum Message <> Reply to Message

Francois writes:

> I am trying to use the iimage tool.
>
> When I start it like this, it works fine:
> 1) IDL> iimage
> 2) browse and open my file.
>
> When I launch it in my code, then it jams:
> 1) ar = read_tiff('C:\image.tif')
> 2) iimage, ar
>
> Is this normal ?

Perfectly normal. You need to file down the little notch on
the door that allows the image to be loaded. It sticks up
too far in the production version.

Cheers,

David

P.S. We might just need a *couple* of more details to
solve this particular problem. :-)

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: iimage tool
Posted by Francois on Tue, 15 Jun 2004 15:14:04 GMT
View Forum Message <> Reply to Message

Ok.
What about the *couple* of more details ?

In the IDL 6.0 Reference Guide (page 892), it gives an example, like mine.
Still doesn't work.
The same example is given in the iimage help topic:

```
file = FILEPATH('mineral.png', SUBDIRECTORY = ['examples', 'data'])
data = READ_PNG(file)
IIMAGE, data, TITLE = 'Electron Image of Mineral Deposits'
```

Thanks,

"David Fanning" <davidf@dfanning.com> wrote in message
news:MPG.1b38b71fb7cefe2f98979f@news.frii.com...
> Francois writes:
>
>> I am trying to use the iimage tool.
>>
>> When I start it like this, it works fine:
>> 1) IDL> iimage
>> 2) browse and open my file.
>>
>> When I launch it in my code, then it jams:
>> 1) ar = read_tiff('C:\image.tif')
>> 2) iimage, ar
>>
>> Is this normal ?
>
> Perfectly normal. You need to file down the little notch on
> the door that allows the image to be loaded. It sticks up
> too far in the production version.
>
> Cheers,
>
> David
>
> P.S. We might just need a *couple* of more details to
> solve this particular problem. :-)
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: iimage tool
Posted by Paul Van Delst[1] on Tue, 15 Jun 2004 15:54:37 GMT
View Forum Message <> Reply to Message

Francois wrote:
> Ok.
> What about the *couple* of more details ?
>

> In the IDL 6.0 Reference Guide (page 892), it gives an example, like mine.
> Still doesn't work.
> The same example is given in the iimage help topic:
> file = FILEPATH('mineral.png', SUBDIRECTORY = ['examples', 'data'])
> data = READ_PNG(file)
> IIMAGE, data, TITLE = 'Electron Image of Mineral Deposits'

I've never used iImage, but when I used the iSurface tool, I could walk to the local coffee shop and back before the thing was displayed (that's an exaggeration, but I hope you get my point :o). Maybe your problem is similar? What happens if you use iImage as above, and then go to lunch? Is the image displayed upon your return?

paulv

>
> Thanks,
>
> Franï¿½ois.
>
>
>
> "David Fanning" <davidf@dfanning.com> wrote in message
> news:MPG.1b38b71fb7cefe2f98979f@news.frii.com...
>
>> Francois writes:
>>
>>
>>> I am trying to use the iimage tool.
>>>
>>> When I start it like this, it works fine:
>>> 1) IDL> iimage
>>> 2) browse and open my file.
>>>
>>> When I launch it in my code, then it jams:
>>> 1) ar = read_tiff('C:\image.tif')
>>> 2) iimage, ar
>>>
>>> Is this normal ?
>>
>> Perfectly normal. You need to file down the little notch on
>> the door that allows the image to be loaded. It sticks up
>> too far in the production version.
>>
>> Cheers,
>>
>> David
>>

>> P.S. We might just need a *couple* of more details to
>> solve this particular problem. :-)
>>
>> --
>> David Fanning, Ph.D.
>> Fanning Software Consulting, Inc.
>> Coyote's Guide to IDL Programming: http://www.dfanning.com/
>
>
>

Subject: Re: iimage tool
Posted by David Fanning on Tue, 15 Jun 2004 15:59:25 GMT
View Forum Message <> Reply to Message

Francois writes:

> In the IDL 6.0 Reference Guide (page 892), it gives an example, like mine.
> Still doesn't work.
> The same example is given in the iimage help topic:
> file = FILEPATH('mineral.png', SUBDIRECTORY = ['examples', 'data'])
> data = READ_PNG(file)
> IIMAGE, data, TITLE = 'Electron Image of Mineral Deposits'

I don't know. Maybe something is not set-up correctly
in your distribution. This works for me every time.

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: iimage tool
Posted by Francois on Tue, 15 Jun 2004 19:36:02 GMT
View Forum Message <> Reply to Message

Hello,

I think I found the anwser.

At the command prompt, this works fine:

> im = read_tiff('C:\toto.tif')
> iimage, im

Included in my code, it does not work because of breakpoints. The time iimage is started, the processing of the code continues and reaches a breakpoint. Then it jams...

Any idea about how to wait iimage has finish before keeping running the code ?

Thank you,

"Francois" <leduc_francois@hotmail.com> wrote in message news:1087312505.266740@news.drenet.dnd.ca...
> Ok.
> What about the *couple* of more details ?
>
> In the IDL 6.0 Reference Guide (page 892), it gives an example, like mine.
> Still doesn't work.
> The same example is given in the iimage help topic:
> file = FILEPATH('mineral.png', SUBDIRECTORY = ['examples', 'data'])
> data = READ_PNG(file)
> IIMAGE, data, TITLE = 'Electron Image of Mineral Deposits'
>
> Thanks,
>

>
>
>
> "David Fanning" <davidf@dfanning.com> wrote in message
> news:MPG.1b38b71fb7cefe2f98979f@news.frii.com...
>> Francois writes:
>>
>>> I am trying to use the iimage tool.
>>>
>>> When I start it like this, it works fine:
>>> 1) IDL> iimage
>>> 2) browse and open my file.
>>>
>>> When I launch it in my code, then it jams:
>>> 1) ar = read_tiff('C:\image.tif')
>>> 2) iimage, ar
>>>
>>> Is this normal ?

>>
>> Perfectly normal. You need to file down the little notch on
>> the door that allows the image to be loaded. It sticks up
>> too far in the production version.
>>
>> Cheers,
>>
>> David
>>
>> P.S. We might just need a *couple* of more details to
>> solve this particular problem. :-)
>>
>> --
>> David Fanning, Ph.D.
>> Fanning Software Consulting, Inc.
>> Coyote's Guide to IDL Programming: http://www.dfanning.com/
>
>

Subject: Re: iimage tool
Posted by David Fanning on Tue, 15 Jun 2004 20:04:47 GMT
View Forum Message <> Reply to Message

Francois writes:

> I think I found the anwser.
>
> At the command prompt, this works fine:
>> im = read_tiff('C:\toto.tif')
>> iimage, im
>
> Included in my code, it does not work because of breakpoints. The time
> iimage is started, the processing of the code continues and reaches a
> breakpoint. Then it jams...
>
> Any idea about how to wait iimage has finish before keeping running the code
> ?

Breakpoints!? Oh, dear. Hope this is not production
code. :-(

I think I would try removing all your breakpoints,
or at least putting them into the code *after* the
call to iImage. :-)

Are you using breakpoints for something other than debugging
code?

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: iimage tool
Posted by Francois on Wed, 16 Jun 2004 12:32:19 GMT
View Forum Message <> Reply to Message

Hello,

Yes. Breakpoints are only there for debugging. But even without breakpoints,
it jams...
In the following code:

```
pro testing
  im = dist(512)
  iimage, im
  t=1
  print, 'hello'
end
```

the image im is being displayed after the "print hello". It is not normal.
The execution of the code should pause while iimage is running.
Like using XROI...

So if I put a breakpoint at line "t=1", then iimage jams. Nothing happens.
And the breakpoint is after the call of iimage...

Have a nice day.

Francois.

P.S. I have sent the question to IDL suppport.

"David Fanning" <davidf@dfanning.com> wrote in message
news:MPG.1b3904455205aefc9897a1@news.frii.com...
> Francois writes:
>
>> I think I found the anwser.
>>
>> At the command prompt, this works fine:

>>> im = read_tiff('C:\toto.tif')
>>> iimage, im
>>
>> Included in my code, it does not work because of breakpoints. The time
>> iimage is started, the processing of the code continues and reaches a
>> breakpoint. Then it jams...
>>
>> Any idea about how to wait iimage has finish before keeping running the
code
>> ?
>
> Breakpoints!? Oh, dear. Hope this is not production
> code. :-(
>
> I think I would try removing all your breakpoints,
> or at least putting them into the code *after* the
> call to iImage. :-)
>
> Are you using breakpoints for something other than debugging
> code?
>
> Cheers,
>
> David
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: iimage tool
Posted by David Fanning on Wed, 16 Jun 2004 13:54:48 GMT
View Forum Message <> Reply to Message

Francois writes:

> Yes. Breakpoints are only there for debugging. But even without breakpoints,
> it jams...
> In the following code:
>   pro testing
>   im = dist(512)
>   iimage, im
>   t=1
>   print, 'hello'
>   end
> the image im is being displayed after the "print hello". It is not normal.

> The execution of the code should pause while iimage is running.
> Like using XROI...

No, iTools are always non-blocking widgets, as far as I can
tell. I see no keyword that can be set to make it block.

> So if I put a breakpoint at line "t=1", then iimage jams. Nothing happens.
> And the breakpoint is after the call of iimage...

Oh, that is odd. You're right.

> P.S. I have sent the question to IDL suppport.

Wonderful. Please let us know what they have to say about
this. :-)

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: iimage tool
Posted by Francois on Wed, 23 Jun 2004 15:50:54 GMT
View Forum Message <> Reply to Message

Hello,

Concerning the iimage command, here is the answer from RSI.

> If you call an iTool from within a program, the iTool will not start
responding to events until focus is returned to the Main level (The IDL
> command line). You can get around this by periodically checking for events
using the WIDGET_EVENT routine. The following example demonstrates how this

> can be done:

```
;*************************************************

pro test_widget_event, check_events = check_events

iimage, dist(300)

check = keyword_set(check_events)
```

```
for i = 0, 100 do begin

print, i

wait, 0.1

if check then $

void = widget_event(/nowait)

endfor

end

;***********************************************
```

Run the program with the CHECK_EVENTS keyword to have the program check for

events from the iTool while in the FOR..DO loop.

Cheers,

Daryl Atencio

Technical Support Engineer

Research Systems, Inc. A Kodak Company

Subject: Re: iimage tool
Posted by David Fanning on Wed, 23 Jun 2004 16:13:52 GMT
View Forum Message <> Reply to Message

Francois writes:

> Concerning the iimage command, here is the answer from RSI.
>
>
> If you call an iTool from within a program, the iTool will not start
> responding to events until focus is returned to the Main level (The IDL
> command line). You can get around this by periodically checking for events
> using the WIDGET_EVENT routine. The following example demonstrates how this
> can be done:

Oh, dear. No debugging allowed. Silent error handlers...

Do you get the impression we aren't suppose to be fooling
around with this software!? :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: iimage tool
Posted by Mark Hadfield on Thu, 24 Jun 2004 01:28:27 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
> Francois writes:
>
>
>> Concerning the iimage command, here is the answer from RSI.
>>
>>
>> If you call an iTool from within a program, the iTool will not start
>> responding to events until focus is returned to the Main level (The IDL
>> command line). You can get around this by periodically checking for events
>> using the WIDGET_EVENT routine. The following example demonstrates how this
>> can be done:
>
>
> Oh, dear. No debugging allowed. Silent error handlers...
>
> Do you get the impression we aren't suppose to be fooling
> around with this software!? :-)

Have you noticed that FSC_SURFACE exhibits the behaviour? :-)

The no-event-handling-while-waiting-at-a-breakpoint issue applies to all (non-blocking) widget programs. Some will display a graph in this situation, some will show an empty window (probably waiting for an expose event), but in all cases you can't interact with them because the widget queue is stopped.

This has frustrated me for some time. When I stop a program at a breakpoint I want to be able to make use of the full range of visualisation tools, not just blocking widgets and non-widget commands.

I recall this being discussed on the group, but I don't recall any simple workaround being offered. (I vaguely recall there might have been one, but I didn't pay enough attention to the thread, and a Google search right now hasn't found anything.)

Based on Daryl Attencio's code, I came up with the routine below (currently called MGH_YIELD, but I'm sure there's a better name).

The idea is that you have stopped at a breakpoint and you want to use IPLOT (or FSC_SURFACE or whatever) to look at the variables. Or, as in Francois's case, your code has called one of these routines before the breakpoint.) So you launch IPLOT (if it's not already active) then type MGH_YIELD at the IDL prompt. The command-line goes grey but you can manipulate your widgets to your heart's content. When you want to recover the command line, you go back to IDLDE and press Ctrl-Break (on Windows) to interrupt MGH_YIELD. This dumps you in MGH_YIELD, whence you can use Ctrl-UpArrow, or "return" at the IDL prompt, to get back to the breakpoint. (This is the fragile part of the whole procedure. It's easy to get lost in the call stack--you can use "help, /TRACEBACK" to check where you are.)

This seems to work robustly under IDLDE in Windows

It might well be that the widget loop in MGH_YIELD could be wrapped in a simple widget application with an Interrupt button, so that control returns to the point where the application was launched.

--
Mark Hadfield "Ka puwaha te tai nei, Hoea tatou"
m.hadfield@niwa.co.nz
National Institute for Water and Atmospheric Research (NIWA)

```
pro MGH_YIELD
   while 1B do begin
      wait, 0.1
      void = widget_event(/NOWAIT)
   endwhile
end
```