Subject: Re: Is that window open?? Posted by hegde on Wed, 02 Apr 1997 08:00:00 GMT

View Forum Message <> Reply to Message

In article <33427754.41C6@irc.chmcc.org>, Phil Williams <williams@irc.chmcc.org> writes:

- > How would I check to see if a window ID is still open in a widget
- > program?

>

- > Here's what I did
- > oldwin = !d.window
- > wset, info.wid

>

> : some code here

>

- > ;--- restore the oldwin
- > wset, oldwin

>

- > and it crashes with
- > % WSET: Window is closed and unavailable.

Use CATCH to trap the error signal and act accordingly. For example:

oldwin = !d.window WSET, info.wid

; some code here

CATCH, err\_status
CASE err\_status OF
-324 : BEGIN
; do what you want when the window is closed
PRINT, 'warning .......'
END

ELSE: WSET, info.wid ENDCASE

CATCH, err status, /CANCEL

As in any other structured language, IDL doesn't define these error signals as constants. Hence programmer is at a disadvantage, if RSI changes them with each version.

Hope it helps.

-M. Hegde

Subject: Re: Is that window open?? Posted by David Foster on Wed, 02 Apr 1997 08:00:00 GMT

View Forum Message <> Reply to Message

```
Phil Williams wrote:
> How would I check to see if a window ID is still open in a widget
> program?
>
> Here's what I did
> oldwin = !d.window
> wset, info.wid
> ; some code here
> ;--- restore the oldwin
> wset, oldwin
> and it crashes with
> % WSET: Window is closed and unavailable.
I think the most straightforward thing to do would be:
oldwin = !d.window
device, window state=windows
case (windows(info.wid)) of
 0: <window is closed...recreate or whatever>
 1: wset. info.wid
endcase
Dave
  David S. Foster
                      Univ. of California, San Diego
  Programmer/Analyst Brain Image Analysis Laboratory
  foster@bial1.ucsd.edu Department of Psychiatry
  (619) 622-5892
                       8950 Via La Jolla Drive, Suite 2200
                 La Jolla, CA 92037
                 [ UCSD Mail Code 0949 ]
```

Subject: Re: Is that window open??

Posted by Tim Patterson on Wed, 02 Apr 1997 08:00:00 GMT

View Forum Message <> Reply to Message

Sorry if I'm on the wrong track, but wouldn't a window ID

being closed ina widget program also imply that the container widget has disappeared too? Couldn't you check for the widget existing still (I presume you're worried about the user closing the window using the Window manager menu rather than letting the program close it).

Tim

```
Phil Williams wrote:
```

>

- > How would I check to see if a window ID is still open in a widget
- > program?

>

- > Here's what I did
- > oldwin = !d.window
- > wset, info.wid

>

> ; some code here

>

- > ;--- restore the oldwin
- > wset, oldwin

>

- > and it crashes with
- > % WSET: Window is closed and unavailable.

> Thanks,

> Phil

Subject: Re: Is that window open??

Posted by brian.jackel on Wed, 02 Apr 1997 08:00:00 GMT

View Forum Message <> Reply to Message

In article <33427754.41C6@irc.chmcc.org> Phil Williams <williams@irc.chmcc.org> writes:

- > How would I check to see if a window ID is still open in a widget
- > program?

For the "standard" set of windows, DEVICE, WINDOW\_STATE=window\_state works, but this seems to apply only to windows 0 to 30, and the Draw\_widget comes out of a pool from 31 or more. I ran across this when trying to make a window manager which would remember plotting states in different windows, but never found a good way to handle widget windows. In the end I wrapped the WSET inside an EXECUTE(), and caught the error (if any) that way. Please post if some more graceful approach is possible.

Subject: Re: Is that window open??

Posted by David Foster on Thu, 03 Apr 1997 08:00:00 GMT

View Forum Message <> Reply to Message

## Brian Jackel wrote:

>

> In article <33427754.41C6@irc.chmcc.org> Phil Williams <williams@irc.chmcc.org> writes:

>

>> How would I check to see if a window ID is still open in a widget

>> program?

>

- > For the "standard" set of windows, DEVICE, WINDOW STATE=window state
- > works, but this seems to apply only to windows 0 to 30, and the Draw\_widget
- > comes out of a pool from 31 or more. I ran across this when trying to make
- > a window manager which would remember plotting states in different windows,
- > but never found a good way to handle widget windows.

I don't think this is correct, at least on my system which is a Sparc 2 running IDL 4.0.1. Check out the following session:

IDL> device, window state=win

IDL> help,win

WIN BYTE = Array[36]

IDL> for i = 0, 50 do window, xsize=10, ysize=10, /free

IDL> device, window\_state=win

IDL> help, win

WIN BYTE = Array[83]

IDL>

Now, win(0:31)=0, win(32:82)=1

I've used WINDOW\_STATE to manage windows in draw-widgets within a "show-image" application where we have over 100 windows up at one time.

Dave

--

David S. Foster Univ. of California, San Diego Programmer/Analyst Brain Image Analysis Laboratory foster@bial1.ucsd.edu Department of Psychiatry (619) 622-5892 8950 Via La Jolla Drive, Suite 2200 La Jolla, CA 92037

Subject: Re: Is that window open??

Posted by steinhh on Thu, 03 Apr 1997 08:00:00 GMT

View Forum Message <> Reply to Message

In article <bri> sprian.jackel.115.5380EC62@uwo.ca>, brian.jackel@uwo.ca (Brian Jackel) writes:

[..]
|> For the "standard" set of windows, DEVICE,WINDOW\_STATE=window\_state
|> works, but this seems to apply only to windows 0 to 30, and the Draw\_widget
|> comes out of a pool from 31 or more. I ran across this when trying to make
|> a window manager which would remember plotting states in different windows,
|> but never found a good way to handle widget windows. In the end I wrapped
|> the WSET inside an EXECUTE(), and caught the error (if any) that way. Please
|> post if some more graceful approach is possible.

I don't think this is a problem - the array size returned by the WINDOW\_STATE keyword expands as you go (at least for me), e.g.:

IDL> device,window\_state=w
IDL> help,w
W BYTE = Array(32)
IDL> base=widget\_base()
IDL> drw=widget\_draw(base,xsize=10,ysize=10)
IDL> widget\_control,base,/realize
IDL> device,window\_state=ww
IDL> help,ww
WW BYTE = Array(33)

The reason I know is that I once made a system for saving/restoring plot region states which works OK. It's available as a tiny part of the solarsoft library (see e.g., http://www.space.lockheed.com/eit/ssw\_setup.html)

If you don't want the whole package, the routines are called pstore.pro, prestore.pro, pconvert.pro, pfind.pro, at http://sohowww.nascom.nasa.gov/solarsoft/gen/idl/display/

There may be dependencies on other software in the solarsoft/gen tree, though... at least the following:

```
../idl/string/trim.pro
../idl/system/get_viewport.pro
../idl/system/setscale.pro
```

../idl/system/setwindow.pro Regards,

Stein Vidar