
Subject: widget_text

Posted by [joerg.plewe](#) on Mon, 25 Apr 1994 13:22:22 GMT

[View Forum Message](#) <> [Reply to Message](#)

idlusers-news-gateway:

id AA09194; Mon, 25 Apr 94 15:22:27 +0200

id AA20146; Mon, 25 Apr 94 15:21:48 +0200

id AA08132; Mon, 25 Apr 94 15:21:37 +0200

Hello!

We are trying to run IDL 3.5.1 under Solaris 2.3.

We want to execute applications written with IDL 3.1.0 and SunOS 4.

Now we have difficulties with WIDGET_TEXTs, when the VALUE is set to "".

```
pro t
```

```
  b=widget_base(Title="Hallo",/column)
```

```
  t=widget_text(b, VALUE="", /editable)
```

```
  t=widget_text(b, VALUE="", /editable)
```

```
  widget_control, b,/realize
```

```
  Xmanager, "t",b
```

```
end
```

We receive the following error message:

```
% X windows protocol error: (BadPixmap).
```

There are a lot of widget_texts, and after a while, the application crashes.

pickfile() and others also seem to use empty widget_text, so that changing our source is not a complete solution!!

With hope for help,

- Joerg Plewe, MPI-Dortmund, Germany

Subject: Re: widget_text

Posted by [davidf](#) on Wed, 10 Jun 1998 07:00:00 GMT

Lisa Bryan (lbryan@arete-az.com) writes:

> I've been experimenting with WIDGET_TEXT hoping to make a text box
> which will show the user messages from anywhere in my application.

...[nice solution snipped]...

> Is there a better way to avoid this work around?

I don't think that is a work-around. I think you found the only real solution.

> P.S. Is there really no way I can get rid of the frame around a text
> box under Windows!?!?

Not that I know of.

Nice programming style, by the way. :-)

Cheers,

David

David Fanning, Ph.D.
Fanning Software Consulting
E-Mail: davidf@dfanning.com
Phone: 970-221-0438
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Subject: Re: widget_text
Posted by [davidf](#) on Fri, 12 Jun 1998 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

J.D. Smith (jdsmith@astrosun.tn.cornell.edu) writes:

> Come now David, there's always a way!

Oh, I didn't argue there wasn't a *way*. But Lisa asked for a "better way". I have to admit that I'm still not sure this qualifies under that criterion. Although I will say in its favor that I learned at least two new text widget keywords and at least one more caveat with respect to "cross-platform" compatibility. :-)

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting
E-Mail: davidf@dfanning.com
Phone: 970-221-0438
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Subject: Re: widget_text
Posted by [J.D. Smith](#) on Fri, 12 Jun 1998 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning wrote:

>
> Lisa Bryan (lbryan@arete-az.com) writes:
>
>> I've been experimenting with WIDGET_TEXT hoping to make a text box
>> which will show the user messages from anywhere in my application.
>
> ...[nice solution snipped]...
>
>> Is there a better way to avoid this work around?
>
> I don't think that is a work-around. I think you found
> the only real solution.
>
>> P.S. Is there really no way I can get rid of the frame around a text
>> box under Windows!?!
>
> Not that I know of.
>
> Nice programming style, by the way. :-)
>
> Cheers,
>
> David

Come now David, there's always a way!

How about a convenience routine such as:

```
pro add_message, state_ptr, message
  widget_control, (*state_ptr).text_box, set_value=message, /append

;; Find the last line in the text box
```

```

nt=widget_info((*state_ptr).text_box,/TEXT_NUMBER)
last_line=(widget_info((*state_ptr).text_box,TEXT_OFFSET_TO_XY=nt-1))[1]

;; Only adjust the scrolling if we've scrolled off the bottom
if last_line lt (*state_ptr).box_ysize then return
widget_control, (*state_ptr).text_box, set_text_top_line = $
    last_line-(*state_ptr).box_ysize+1
end

```

Note that this does *not* work with text widgets with /WRAP set (though see below), and that another state variable, box_ysize (the ysize of the text widget), in lieu of your top_line variable, is required (...well, not strictly, since you could do a widget_info(/GEOMETRY,box) to find out the box's ysize -- but that's pretty inefficient). It will still work even if some inconsiderate application or user adds text without doing it nicely (i.e. incrementing your top_line variable).

Or, for an entirely magic way, at least on Linux, just set /WRAP on your text widget when created. That gives you this behaviour automatically... i.e. you don't have to go through the set_text_top_line business (in fact, that would fail), and the call:

```

widget_control, (*state_ptr).text_box, set_value = 'my text', /append

```

will do this all by itself. I'm not sure if this /WRAP solution works for all platforms, but it certainly is the easiest way. Just tested and it does not work on Windows v5.0.

Good Luck,

JD

--

```

J.D. Smith                [*]    WORK: (607) 255-5842
Cornell University Dept. of Astronomy [*]    (607) 255-4083
206 Space Sciences Bldg.    [*]    FAX: (607) 255-5875
Ithaca, NY 14853          [*]

```

Subject: Re: widget_text
 Posted by [davidf](#) on Sat, 13 Jun 1998 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

J.D. Smith (jdsmith@astrosun.tn.cornell.edu) writes:

- > But the real advantage of this
- > method is that it works without having to trust a state variable line
- > count (since the widget itself is queried for the number of lines it
- > contains). It is therefore well suited to the creation of a single text
- > box for messages originating from mostly unrelated applications (sort of

- > like a console), without each application having to worry about what the
- > others are up to. And furthermore, we've got to exercise all of those
- > various keywords to make sure RSI isn't just fattening up the manual
- > with made up material!

Well, you're right. Putting this kind of functionality in a procedure is a good idea. And I say the more keywords there are, the merrier, too. Were you around J.D. when there were only 11 keywords to Widget_Control? Boy, those were the good old days. You could learn how to write a widget program in about half an hour. Although it took most of us nearly a year to screw up our courage enough to try it the first time. We were the 60's generation, I guess. :-)

Cheers,

David

P.S. I've decided to spend two or three days working my way through text widget keywords. I didn't realize there was so much there. :-(

--

David Fanning, Ph.D.

Fanning Software Consulting

E-Mail: davidf@dfanning.com

Phone: 970-221-0438

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Subject: Re: widget_text

Posted by [J.D. Smith](#) on Sat, 13 Jun 1998 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

David Fanning wrote:

- >
- > J.D. Smith (jdsmith@astrosun.tn.cornell.edu) writes:
- >
- >> Come now David, there's always a way!
- >
- > Oh, I didn't argue there wasn't a *way*. But Lisa asked
- > for a "better way". I have to admit that I'm still not
- > sure this qualifies under that criterion. Although I will
- > say in its favor that I learned at least two new
- > text widget keywords and at least one more caveat
- > with respect to "cross-platform" compatibility. :-)
- >
- > Cheers,
- >

> David
>

And I thought Lisa's issue was she didn't want to go through the mess of

```
widget_control,...,/APPEND  
widget_control,...,SET_TEXT_TOP_LINE=...
```

every time she wanted to add a message line... not that packaging it in a procedure is an unheard of trick... But the real advantage of this method is that it works without having to trust a state variable line count (since the widget itself is queried for the number of lines it contains). It is therefore well suited to the creation of a single text box for messages originating from mostly unrelated applications (sort of like a console), without each application having to worry about what the others are up to. And furthermore, we've got to exercise all of those various keywords to make sure RSI isn't just fattening up the manual with made up material!

JD

--

J.D. Smith |*| WORK: (607) 255-5842
Cornell University Dept. of Astronomy |*| (607) 255-4083
206 Space Sciences Bldg. |*| FAX: (607) 255-5875
Ithaca, NY 14853 |*|

Subject: Re: widget_text
Posted by [Phillip & Suzanne](#) on Thu, 18 Jun 1998 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning wrote:

> Well, you're right. Putting this kind of functionality
> in a procedure in a good idea.

David, Lisa, and all;

I agree with the sentiment, but thought of another possibility, with IDL 5 having objects and all...

Make a message object, which contains a text widget. It could have an append method that takes a text string, and performs all of the steps you mention. In this fashion, you would not have to add all of this code everywhere. Instead, you would just write something like:

```
(*state_ptr).message->append('my text')
```

where the append method would do all of the steps you were doing.

I realize that this is a similar solution to making a procedure for it, but it allows the whole thing to be encapsulated (in good OO fashion), and would allow the top line to be preserved as part of the object's data structure. Without real effort, no one would be able to add DIRECTLY to the widget, because it's encapsulated within a message object. Also, if you decided to change the implementation of the message object some day (maybe when RSI introduces another widget even better designed for this sort of purpose? :-), changing the internals of your message object wouldn't affect the rest of your program.

A little food for thought...

Phillip David

Subject: Re: widget_text

Posted by [David Fanning](#) on Thu, 21 Feb 2002 16:42:56 GMT

[View Forum Message](#) <> [Reply to Message](#)

John Copley (john.copley@nist.gov) writes:

> It seems that widget_text cannot handle reasonably large string arrays
> and I'm wondering if anyone can suggest a workaround. I'd like to be
> able to display string arrays with about 100,000 characters total.
>

> Below are some lines of code where I generate an array of 1000
> strings, each of which is 12*n characters in length; within a loop n
> ranges from 1 to 10. I then seek to display the string array using
> widget_text. For n=1-3 it's OK but for larger n the number of lines
> displayed goes as follows: (n is in the first column and the number of
> lines is in the last column). (This is the output from the code)

>	1	12	12	14000	14000.0	1000
>	2	24	24	26000	26000.0	1000
>	3	36	36	38000	38000.0	1000
>	4	48	48	33380	50000.0	667
>	5	60	60	41136	62000.0	663
>	6	72	72	50868	74000.0	687
>	7	84	84	31936	86000.0	371
>	8	96	96	57752	98000.0	589
>	9	108	108	53160	110000.	483
>	10	120	120	35288	122000.	289

Humm. Here is what I get when I run the program in IDL 5.5 windows:

IDL> test_xdisplayfile

1	12	12	14000	14000.0	1000
---	----	----	-------	---------	------

2	24	24	26000	26000.0	1000
3	36	36	38000	38000.0	1000
4	48	48	50000	50000.0	1000
5	60	60	62000	62000.0	1000
6	72	72	74000	74000.0	1000
7	84	84	86000	86000.0	1000
8	96	96	98000	98000.0	1000
9	108	108	110000	110000.	1000
10	120	120	122000	122000.	1000

Cheers,

David

--

David W. Fanning, Ph.D.

Fanning Software Consulting

Phone: 970-221-0438, E-mail: david@dfanning.com

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Toll-Free IDL Book Orders: 1-888-461-0155
