
Subject: Predicting string lengths in widgets
Posted by [mattf](#) on Mon, 06 Aug 2007 11:41:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi all...

I have a table widget and I'd like to set the column widths -before- the table is realized so that the user doesn't have to fiddle with adjustable widths. Unfortunately, I don't know the lengths of the strings that I'm going to put into the table before running the program, and the strings may be rather long. And, for a cross-platform application, I don't know what the font will be, or, for that matter, the size of the display.

At the moment, I'm just hacking at the problem by using a pixmap window and the width parameter in xyouts to get an idea of the string length in pixels, e.g., if the string is 'test_str' I do

```
str_max_sz  
window, /free, /pixmap, xsize = str_max_sz, ysize = str_max_sz  
xyouts, test_str, width = w  
wdelete  
str_pix_len = ceil(w*str_max_sz)
```

and then use a fudge factor to get good-looking column widths from str_pix_len. But there must be a way of doing it, you know, correctly... no?

Subject: Re: Predicting string lengths in widgets
Posted by [Michael Galloy](#) on Mon, 06 Aug 2007 16:40:30 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Aug 6, 5:41 am, mattf <mfe...@earthlink.net> wrote:

> Hi all...

>

> I have a table widget and I'd like to set the column widths -before-
> the table is realized so that the user doesn't have to fiddle with
> adjustable widths. Unfortunately, I don't know the lengths of the
> strings that I'm going to put into the table before running the
> program, and the strings may be rather long. And, for a cross-platform
> application, I don't know what the font will be, or, for that matter,
> the size of the display.

>

> At the moment, I'm just hacking at the problem by using a pixmap
> window and the width parameter in xyouts to get an idea of the string
> length in pixels, e.g., if the string is 'test_str' I do

>

```
> str_max_sz
> window, /free, /pixmap, xsize = str_max_sz, ysize = str_max_sz
> xyouts, test_str, width = w
> wdelete
> str_pix_len = ceil(w*str_max_sz)
>
> and then use a fudge factor to get good-looking column widths from
> str_pix_len. But there must be a way of doing it, you know,
> correctly... no?
```

Check out the `STRING_SIZE` keyword to `WIDGET_INFO`. (IDL 6.2+ required.)

Mike

--

www.michaelgalloy.com

Subject: Re: Predicting string lengths in widgets
Posted by [mattf](#) on Mon, 06 Aug 2007 17:08:59 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Aug 6, 12:40 pm, "mgal...@gmail.com" <mgal...@gmail.com> wrote:

```
> On Aug 6, 5:41 am, mattf <mfe...@earthlink.net> wrote:
>
>
>
>> Hi all...
>
>> I have a table widget and I'd like to set the column widths -before-
>> the table is realized so that the user doesn't have to fiddle with
>> adjustable widths. Unfortunately, I don't know the lengths of the
>> strings that I'm going to put into the table before running the
>> program, and the strings may be rather long. And, for a cross-platform
>> application, I don't know what the font will be, or, for that matter,
>> the size of the display.
>
>> At the moment, I'm just hacking at the problem by using a pixmap
>> window and the width parameter in xyouts to get an idea of the string
>> length in pixels, e.g., if the string is 'test_str' I do
>
>> str_max_sz
>> window, /free, /pixmap, xsize = str_max_sz, ysize = str_max_sz
>> xyouts, test_str, width = w
>> wdelete
>> str_pix_len = ceil(w*str_max_sz)
>
>> and then use a fudge factor to get good-looking column widths from
>> str_pix_len. But there must be a way of doing it, you know,
```

>> correctly... no?

>

> Check out the `STRING_SIZE` keyword to `WIDGET_INFO`. (IDL 6.2+ required.)

>

> Mike

> --www.michaelgalloy.com

Thanks. I saw that, actually, but the documentation is mercilessly brief. Does the widget have to be realized? Do all the widgets on a given base inherit the same default string-to-pixels scaling? I suppose I could try applying `WIDGET_INFO` with that keyword to some realized widget in the widget application, and then kinda assume that a string in my table would work the same way... Still kind of a hack, but seems closer to the right answer. Hmm...
