
Subject: Re: Buttons in Widgets

Posted by [lecacheux.alain](#) on Mon, 25 Jun 2012 11:17:47 GMT

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

On 25 juin, 11:54, Helder <hel...@marchetto.de> wrote:

> Hi,

> I'm don't like that much how the buttons of IDL widgets look (Widget_button) even when using the /bitmap argument. I am exploring the solution of using a draw_widget, making it of the same color as the background(*) and then filling it is with my own drawing. The difference between the two methods is that the borders of the buttons will appear different: for a button widget these are defined by the IDL and for a draw widget one can have "no" borders or define how these look like. Another advantage is that one can have the effect of having an something like an icon floating on a background.

> Has anybody tried doing such things? So far it seems to do the job for me, but I haven't played too much with it yet.

> (*) Is there a way to find out what color the base widget has and/or to define it?

>

> Cheers,

> Helder

>

>

You might be aware that IDL 8.2 has enhancement for Widget_Button (color bitmap and shape). I have not used so far, but you should have a look.
alain.

Subject: Re: Buttons in Widgets

Posted by [Helder Marchetto](#) on Mon, 25 Jun 2012 11:48:26 GMT

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

On Monday, June 25, 2012 1:17:47 PM UTC+2, alx wrote:

> On 25 juin, 11:54, Helder <hel...@marchetto.de> wrote:

>> Hi,

>> I'm don't like that much how the buttons of IDL widgets look (Widget_button) even when using the /bitmap argument. I am exploring the solution of using a draw_widget, making it of the same color as the background(*) and then filling it is with my own drawing. The difference between the two methods is that the borders of the buttons will appear different: for a button widget these are defined by the IDL and for a draw widget one can have "no" borders or define how these look like. Another advantage is that one can have the effect of having an something like an icon floating on a background.

>> Has anybody tried doing such things? So far it seems to do the job for me, but I haven't played too much with it yet.

>> (*) Is there a way to find out what color the base widget has and/or to define it?

>>

>> Cheers,

>> Helder

>>
>>
>
> You might be aware that IDL 8.2 has enhancement for Widget_Button
> (color bitmap and shape). I have not used so far, but you should have
> a look.
> alain.

Hi Alain,
on the IDL Help I only found that "Widgets on Microsoft Windows oplatforms have a more modern appearance, using rounded edges for bitmap buttons."
Buttons do appear different in 8.2, but I don't yet have the freedom (look) I wished for.
Thanks, Helder

Subject: Re: Buttons in Widgets
Posted by [David Fanning](#) on Mon, 25 Jun 2012 12:36:18 GMT
[View Forum Message](#) <> [Reply to Message](#)

Helder writes:

> I'm don't like that much how the buttons of IDL widgets
> look (Widget_button) even when using the /bitmap argument.
> I am exploring the solution of using a draw_widget, making
> it of the same color as the background(*) and then filling
> it is with my own drawing. The difference between the two
> methods is that the borders of the buttons will appear
> different: for a button widget these are defined by the
> IDL and for a draw widget one can have "no" borders or
> define how these look like. Another advantage is that
> one can have the effect of having an something like
> an icon floating on a background.
> Has anybody tried doing such things?

Of course. :-)

To be "real" for users (i.e., they get the idea they have
"pushed" something) you need to have two states for your
button: a "normal" look and a "pushed" look. Depending
on how you draw your button, this sometimes is just a
matter of reversing some or all of your colors (you need
about 6-10 to make a decent looking button). You definitely
want to build this as a compound object-widget, because
there is going to be a lot of things your button will
want to remember.

In the end, I think you will find that it is probably
just a whole lot easier and more efficient to change

your attitude about WIDGET_BUTTON and use it. :-)

> (*) Is there a way to find out what color the base
> widget has and/or to define it?

Use the System_Colors keyword to Widget_Info to get this information. I carried this information around in cgColor for years, but no one I knew (including me after I gave up on making my own buttons) ever used it, so I removed it. It requires a window connection to obtain these colors, so don't distribute your application widely, unless you wish additional headaches. :-)

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

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

Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Buttons in Widgets

Posted by [lecacheux.alain](#) on Mon, 25 Jun 2012 14:14:10 GMT

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

On 25 juin, 14:36, David Fanning <n...@idlcoyote.com> wrote:

> Helder writes:

>> I'm don't like that much how the buttons of IDL widgets
>> look (Widget_button) even when using the /bitmap argument.
>> I am exploring the solution of using a draw_widget, making
>> it of the same color as the background(*) and then filling
>> it is with my own drawing. The difference between the two
>> methods is that the borders of the buttons will appear
>> different: for a button widget these are defined by the
>> IDL and for a draw widget one can have "no" borders or
>> define how these look like. Another advantage is that
>> one can have the effect of having an something like
>> an icon floating on a background.
>> Has anybody tried doing such things?

>

> Of course. :-)

>

> To be "real" for users (i.e., they get the idea they have

> "pushed" something) you need to have two states for your

> button: a "normal" look and a "pushed" look. Depending
> on how you draw your button, this sometimes is just a
> matter of reversing some or all of your colors (you need
> about 6-10 to make a decent looking button). You definitely
> want to build this as a compound object-widget, because
> there is going to be a lot of things your button will
> want to remember.
>
> In the end, I think you will find that it is probably
> just a whole lot easier and more efficient to change
> your attitude about WIDGET_BUTTON and use it. :-)
>
>> (*) Is there a way to find out what color the base
>> widget has and/or to define it?
>
> Use the System_Colors keyword to Widget_Info to get
> this information. I carried this information around in
> cgColor for years, but no one I knew (including me
> after I gave up on making my own buttons) ever
> used it, so I removed it. It requires a window
> connection to obtain these colors, so don't distribute
> your application widely, unless you wish additional
> headaches. :-)
>
> Cheers,
>
> David
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming: <http://www.idlcoyote.com/>
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")
>
>

Regarding "pushed" appearance, it seems to me only a matter of
exchanging some predefined bitmaps at each push, like by doing:

WIDGET_CONTROL, your_button, SET_VALUE=pushed ? OnBitmap :
OffBitmap, /BITMAP
keeping you able to use any further widget_button facility you like.
alain.

Subject: Re: Buttons in Widgets
Posted by [David Fanning](#) on Mon, 25 Jun 2012 14:21:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

alx writes:

- > Regarding "pushed" appearance, it seems to me only a matter of
- > exchanging some predefined bitmaps at each push, like by doing:
- > WIDGET_CONTROL, your_button, SET_VALUE=pushed ? OnBitmap :
- > OffBitmap, /BITMAP
- > keeping you able to use any further widget_button facility you like.

I was talking about draw widgets that act as button widgets. Bitmaps were WAY too unreliable when I was making my own buttons. Perhaps they are better now. But, changing widget visual properties on the fly has never been one of IDL's great strengths. Although, I will say this, it usually works on Windows machines. :-)

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

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

Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Buttons in Widgets

Posted by [Michael Galloy](#) on Mon, 25 Jun 2012 16:41:06 GMT

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

On 6/25/12 3:54 AM, Helder wrote:

- > Hi,
- > I'm don't like that much how the buttons of IDL widgets look (Widget_button) even when using the /bitmap argument. I am exploring the solution of using a draw_widget, making it of the same color as the background(*) and then filling it is with my own drawing. The difference between the two methods is that the borders of the buttons will appear different: for a button widget these are defined by the IDL and for a draw widget one can have "no" borders or define how these look like. Another advantage is that one can have the effect of having an something like an icon floating on a background.
- > Has anybody tried doing such things? So far it seems to do the job for me, but I haven't played too much with it yet.
- > (*) Is there a way to find out what color the base widget has and/or to define it?
- >
- > Cheers,
- > Helder
- >

You can't change the colors, but you can query with:

```
IDL> tlb = widget_base()
IDL> colors = widget_info(tlb, /system_colors)
IDL> help, colors
** Structure WIDGET_SYSTEM_COLORS, 25 tags, length=150, data length=150:
  DARK_SHADOW_3D INT      Array[3]
  FACE_3D      INT      Array[3]
  LIGHT_EDGE_3D INT      Array[3]
  LIGHT_3D     INT      Array[3]
  SHADOW_3D    INT      Array[3]
  ACTIVE_BORDER INT      Array[3]
  ACTIVE_CAPTION INT      Array[3]
  APP_WORKSPACE INT      Array[3]
  DESKTOP      INT      Array[3]
  BUTTON_TEXT  INT      Array[3]
  CAPTION_TEXT INT      Array[3]
  GRAY_TEXT    INT      Array[3]
  HIGHLIGHT    INT      Array[3]
  HIGHLIGHT_TEXT INT      Array[3]
  INACTIVE_BORDER INT      Array[3]
  INACTIVE_CAPTION
      INT      Array[3]
  INACTIVE_CAPTION_TEXT
      INT      Array[3]
  TOOLTIP_BK   INT      Array[3]
  TOOLTIP_TEXT INT      Array[3]
  MENU         INT      Array[3]
  MENU_TEXT    INT      Array[3]
  SCROLLBAR    INT      Array[3]
  WINDOW_BK    INT      Array[3]
  WINDOW_FRAME INT      Array[3]
  WINDOW_TEXT  INT      Array[3]
```

Mike

--

Michael Galloy

www.michaelgalloy.com

Modern IDL: A Guide to IDL Programming (<http://modernidl.idldev.com>)

Research Mathematician

Tech-X Corporation

Subject: Re: Buttons in Widgets

Posted by [Russell Ryan](#) on Mon, 25 Jun 2012 17:00:08 GMT

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

On Monday, June 25, 2012 5:54:13 AM UTC-4, Helder wrote:

> Hi,
> I'm don't like that much how the buttons of IDL widgets look (Widget_button) even when using the /bitmap argument. I am exploring the solution of using a draw_widget, making it of the same color as the background(*) and then filling it is with my own drawing. The difference between the two methods is that the borders of the buttons will appear different: for a button widget these are defined by the IDL and for a draw widget one can have "no" borders or define how these look like. Another advantage is that one can have the effect of having an something like an icon floating on a background.
> Has anybody tried doing such things? So far it seems to do the job for me, but I haven't played too much with it yet.
> (*) Is there a way to find out what color the base widget has and/or to define it?
>
> Cheers,
> Helder

It sounds like you're trying to do something like Rob Dimeo did?

http://www.ncnr.nist.gov/staff/dimeo/idl_programs.html

Check out his compound widgets (ie. the screen shot pages).

Russell
