Subject: Re: Keywords/Parameters and Common Blocks
Posted by Pavel Romashkin on Fri, 31 Oct 2003 17:25:18 GMT
View Forum Message <> Reply to Message

Why can't you simply pass the keyword parameters to subroutines? Why do
you want to use common blocks at all?

Pavel

Nuno Oliveira wrote:
>
> Hi.
>
> Sometimes I want use keyword variables (or parameters) of a function or
> procedure in routines not only in the main program (of the
> function/procedure) but also on subroutines (as often happens with routine
> events). Intuitively I put the keyword variables on a block shared with the
> subroutine that I also want to use those variables. But it gives an error;
> that I can't remember now. It forces me to create new variables where I put
> the contents of the keyword variables, and then those variables I created in
> the program I can share with subroutines with common block.
>
> Is there a way that I can achieve this without making as many attributions
> as many keyword variables I want to share in COMMON blocks?

Subject: Re: Keywords/Parameters and Common Blocks
Posted by David Fanning on Fri, 31 Oct 2003 17:53:00 GMT
View Forum Message <> Reply to Message

Nuno Oliveira writes:

> Sometimes I want use keyword variables (or parameters) of a function or
> procedure in routines not only in the main program (of the
> function/procedure) but also on subroutines (as often happens with routine
> events). Intuitively I put the keyword variables on a block shared with the
> subroutine that I also want to use those variables. But it gives an error;
> that I can't remember now. It forces me to create new variables where I put
> the contents of the keyword variables, and then those variables I created in
> the program I can share with subroutines with common block.
>
> Is there a way that I can achieve this without making as many attributions
> as many keyword variables I want to share in COMMON blocks?

I get the impression here that you are writing widget
programs. If so, you do NOT want to be using COMMON blocks. :-)
Use an "info" or "state" structure instead. See any widget
program on my web page for an example of how this is done.

Typically, if you want to collect keywords for routines,
you store them in your info structure so they can be used:

```
  PRO MyProgram, Color=color, Linestyle=linestyle

  IF N_Elements(color) EQ 0 THEN color = 'red'
  IF N_Elements(linestyle) EQ 0 THEN linestyle = 1

  info = { color:color, linestyle:linestyle }
  Widget_Control, tlb, Set_UValue=info

  etc.
  END
```

Then, you can use them in an event handler:

```
  PRO MyProgramEvents, event
  Widget_Control, event.top, Get_UValue=info
  MYPLOT, findgen(11), Color=info.color, LineStyle=info.linestyle
  END
```

Sometimes you want to collect "extra" keywords, or keywords
that you might want to use, but don't want to take the time
to define. Then you might use a pointer to store these "extra"
keywords:

```
  PRO MyProgram, _Extra=extra

  IF N_Elements(extra) EQ 0 THEN $
 extra = Ptr_New(/Allocate_Heap) ELSE $
     extra = Ptr_New(extra)

  info = { extra=extra}
  Widget_Control, tlb, Set_UValue=info

  etc.
  END
```

Then, use them like this:

```
  PRO MyProgramEvents, event
  Widget_Control, event.top, Get_UValue=info
  MYPLOT, findgen(11), _Extra=*info.extra
  END
```

Just be sure you free your pointers in the CLEANUP procedure
you will write for your widget program. :-)

Cheers,

David

--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
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

Subject: Re: Keywords/Parameters and Common Blocks
Posted by Nuno Oliveira on Tue, 04 Nov 2003 10:56:25 GMT
View Forum Message <> Reply to Message

Thank you, David. I'm already making changes in my programs passing the widget variables to the value of the widget base.

Indeed, I'm trying to build a widget application for medical purposes, but I 'm freshman in IDL. So I'm reading the thousands of pages of the tutorials, looking at yours and other sites e looking for information here, and sometimes talking with RSI.

And sometimes happen things just like this. I started to work with widget programs and immediately found that I needed to pass the variables to the event routine, so I looked in the tutorials and found that I could do that with common blocks. And unfortunately I thought the question was solved, till I noticed that my programs were tremendously confused.

So I hope I don't bore you too much in the future, but I'm getting the idea that this group is very useful to me, because some things are not too obvious in the tutorials (and I still didn't touch the Objects)

Cordially,

Nuno.

"David Fanning" <david@dfanning.com> wrote in message
news:MPG.1a0c4f6649abcabc98972a@news.frii.com...
> Nuno Oliveira writes:
>
>
> I get the impression here that you are writing widget
> programs. If so, you do NOT want to be using COMMON blocks. :-)
> Use an "info" or "state" structure instead. See any widget
> program on my web page for an example of how this is done.
>
> Typically, if you want to collect keywords for routines,
> you store them in your info structure so they can be used:
>
>   PRO MyProgram, Color=color, Linestyle=linestyle
>
>   IF N_Elements(color) EQ 0 THEN color = 'red'
>   IF N_Elements(linestyle) EQ 0 THEN linestyle = 1
>
>   info = { color:color, linestyle:linestyle }
>   Widget_Control, tlb, Set_UValue=info
>
>   etc.
>   END
>
> Then, you can use them in an event handler:
>
>   PRO MyProgramEvents, event
>   Widget_Control, event.top, Get_UValue=info
>   MYPLOT, findgen(11), Color=info.color, LineStyle=info.linestyle
>   END
>
> Sometimes you want to collect "extra" keywords, or keywords
> that you might want to use, but don't want to take the time
> to define. Then you might use a pointer to store these "extra"
> keywords:
>
>   PRO MyProgram, _Extra=extra
>
>   IF N_Elements(extra) EQ 0 THEN $
> extra = Ptr_New(/Allocate_Heap) ELSE $
>   extra = Ptr_New(extra)
>
>   info = { extra=extra}

>   Widget_Control, tlb, Set_UValue=info
>
>   etc.
>   END
>
> Then, use them like this:
>
>   PRO MyProgramEvents, event
>   Widget_Control, event.top, Get_UValue=info
>   MYPLOT, findgen(11), _Extra=*info.extra
>   END
>
> Just be sure you free your pointers in the CLEANUP procedure
> you will write for your widget program. :-)
>
> Cheers,
>
> David
>
>
>
> --
> David W. Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> 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
