
Subject: Re: Subclassing from IDLitTool ??

Posted by [David Fanning](#) on Wed, 01 Sep 2004 16:39:26 GMT

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

Antonio Santiago writes:

>
> i'm starting with iTools and i try this example:
> all is fine. A new iTool component is created (empty but ok) with all
> standard functionality.
>
> The "iTool Developer's Guide" say that if i want an iTool without
> standar functionality i need to subclassing from IDLitTool. I try it
> replacing "IDLitToolbase" to "IDLitTool" in file iprobe__define.pro.
>
> The result is an error (in a dialog window) like:
>
> "A system error was... bla, bla, bla"
> "The value of this error was:"
> "Object reference type required in this context"
>
>
> What do i need to do to subclassing from IDLitTool?

I think I ran into this same problem with the beta version.
I reported it, but after a week and a half of constant
effort I finally decided iTools were too advanced for me
and went back to writing programs I could understand.

For what it's worth, I finally concluded this wasn't
possible with the information I was given to work with.
Perhaps you will have better luck. If so, let me know
and I'll write an article.

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

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

Subject: Re: Subclassing from IDLitTool ??

Posted by [Chris Lee](#) on Wed, 01 Sep 2004 17:24:28 GMT

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

In article <4135CD89.4050300@est.fib.upc.es>, "Antonio Santiago" <d6522117@est.fib.upc.es> wrote:

> What do i need to do to subclassing from IDLitTool? Thanks.

IDLitToolbase comes preloaded with all of the menus in the IDL standard iTools. IDLitTool comes with zero, nadda, zip. You have to supply all of the menu items (for example). Then IDLitTool contains the necessary functions to return the object references.

e.g.

```
;Init method
FUNCTION iprobe::Init

    if( self->IDLitTool::Init(_extra=_extra) eq 0)then $
        return, 0
;file
    self->RegisterOperation, 'Exit', 'IDLitopFileExit', $
        IDENTIFIER='File/Exit', /SEPARATOR

;button
    self->RegisterManipulator, 'Arrow', 'IDLitManipArrow', $
        ICON='arrow', /DEFAULT, IDENTIFIER="ARROW", $
        DESCRIPTION='Click to select items, click and drag to select multiple items'

    return, 1

END
```

That gets you onto the first step...

Chris.

Subject: Re: Subclassing from IDLitTool ??
Posted by [David Fanning](#) on Wed, 01 Sep 2004 17:33:08 GMT
[View Forum Message](#) <> [Reply to Message](#)

Christopher Lee writes:

> IDLitToolbase comes preloaded with all of the menus in the IDL
> standard iTools. IDLitTool comes with zero, nadda,
> zip. You have to supply all of the menu items (for example). Then
> IDLitTool contains the necessary functions to return the object
> references.

```
>
> e.g.
>
> ;Init method
> FUNCTION iprobe::Init
>
>   if( self->IDLitTool::Init(_extra=_extra) eq 0)then $
>     return, 0
> ;file
>   self->RegisterOperation, 'Exit', 'IDLitopFileExit', $
>     IDENTIFIER='File/Exit', /SEPARATOR
>
> ;button
>   self->RegisterManipulator, 'Arrow', 'IDLitManipArrow', $
>     ICON='arrow', /DEFAULT, IDENTIFIER="ARROW", $
>     DESCRIPTION='Click to select items, click and drag to select multiple items'
>
>   return, 1
>
> END
>
> That gets you onto the first step...
```

Didn't work for me. I get the same error as before. :-(

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

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

Subject: Re: Subclassing from IDLitTool ??

Posted by [Chris Lee](#) on Wed, 01 Sep 2004 22:35:49 GMT

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

In article <MPG.1b9fb5c11558d7fc989873@news.frii.com>, "David Fanning" <davidf@dfanning.com> wrote:

```
> Didn't work for me. I get the same error as before. :-( Cheers,
> David
```

I put the following code in a fresh file, compile the file, and then run the IPROBE procedure, it works first time. You can even take out the 'file' menu item. It still works for me. I'm not sure it's a bug, but you need at least one manipulator/tool to start the iTool.

If that doesn't work, but the IDLitToolbase version works. Have a look at the IDLitToolbase source. Strip the menus out of there until your left with the minimum that will compile :)

As you said, this isn't possible with the information given in the documentation.

```
{ x86 linux unix linux 6.0 Jun 27 2003 32 64}
```

Chris.

```
;start
FUNCTION iprobe::Init

    if( self->IDLitTool::Init(_extra=_extra) eq 0)then $
        return, 0
;file
    self->RegisterOperation, 'Exit', 'IDLitopFileExit', $
        IDENTIFIER='File/Exit', /SEPARATOR

;button
    self->RegisterManipulator, 'Arrow', 'IDLitManipArrow', $
        ICON='arrow', /DEFAULT, IDENTIFIER="ARROW", $
        DESCRIPTION='Click to select items, click and drag to select multiple items'

    return, 1

END

;Define object struct
PRO iprobe__define
struct = { iprobe, $
    INHERITS IDLitTool $ ; Provides iTool interface
}
END

PRO iprobe

ITREGISTER, "Example iTool", "iprobe"
identifier = IDLITSYS_CREATEETOOL("Example iTool")
END
```

Subject: Re: Subclassing from IDLitTool ??
Posted by [Antonio Santiago](#) on Thu, 02 Sep 2004 06:30:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thanks floks :)

Ok both Christopher and David are right.

Now with IDL 6.0 Christopher's code runs fine but not in IDL 6.1 (this seems a joke) and yes, the documentation about IDLitTool is horrible.

Thanks and bye.

Christopher Lee wrote:

```
> In article <4135CD89.4050300@est.fib.upc.es>, "Antonio Santiago"
> <d6522117@est.fib.upc.es> wrote:
>
>
>
>> What do i need to do to subclassing from IDLitTool? Thanks.
>
>
> IDLitToolbase comes preloaded with all of the menus in the IDL
> standard iTools. IDLitTool comes with zero, nadda,
> zip. You have to supply all of the menu items (for example). Then
> IDLitTool contains the necessary functions to return the object
> references.
>
> e.g.
>
> ;init method
> FUNCTION iprobe::init
>
>   if( self->IDLitTool::init(_extra=_extra) eq 0)then $
>     return, 0
> ;file
>   self->RegisterOperation, 'Exit', 'IDLitopFileExit', $
>     IDENTIFIER='File/Exit', /SEPARATOR
>
> ;button
>   self->RegisterManipulator, 'Arrow', 'IDLitManipArrow', $
>     ICON='arrow', /DEFAULT, IDENTIFIER="ARROW", $
>     DESCRIPTION='Click to select items, click and drag to select multiple items'
>
>   return, 1
>
```

> END
>
> That gets you onto the first step...
>
> Chris.

Subject: Re: Subclassing from IDLitTool ??
Posted by [Chris\[2\]](#) on Wed, 08 Sep 2004 16:35:42 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Antonio & Chris,

There is a bug in framework/idlitcommandbuffer__define.pro, around line 232, where it assumes that you have the Undo/Redo operations. You can work around this by making sure the Undo/Redo operations are registered in your subclass:

```
self->RegisterOperation, 'Undo', $
  'IDLitopUndo', $
  ACCELERATOR='Ctrl+Z', $
  IDENTIFIER='Edit/Undo', ICON='undo', /disable, $
  /IGNORE_AVAILABILITY
```

```
self->RegisterOperation, 'Redo', $
  'IDLitopRedo', $
  ACCELERATOR='Ctrl+Y', $
  IDENTIFIER='Edit/Redo', ICON='redo', /disable, $
  /IGNORE_AVAILABILITY
```

Or, if you are brave or don't want to have these operations, you can modify idlitcommandbuffer__define.pro by adding a check after line 232:

change this:

```
oUndo = self._oEnv->GetByIdentifier('OPERATIONS/EDIT/UNDO')
idUndo = oUndo->GetFullIdentifier()
```

to the following:

```
oUndo = self._oEnv->GetByIdentifier('OPERATIONS/EDIT/UNDO')
if (~OBJ_VALID(oUndo)) then $
  return
idUndo = oUndo->GetFullIdentifier()
```

-Chris
Research Systems, Inc.

"Christopher Lee" <cl@127.0.0.1> wrote in message
news:20040901.182404.1960709859.9943@buckley.atm.ox.ac.uk...

> In article <4135CD89.4050300@est.fib.upc.es>, "Antonio Santiago"
> <d6522117@est.fib.upc.es> wrote:
>
>
>> What do i need to do to subclassing from IDLitTool? Thanks.
>
> IDLitToolbase comes preloaded with all of the menus in the IDL
> standard iTools. IDLitTool comes with zero, nadda,
> zip. You have to supply all of the menu items (for example). Then
> IDLitTool contains the necessary functions to return the object
> references.
>
> e.g.
>
> ;Init method
> FUNCTION iprobe::Init
>
> if(self->IDLitTool::Init(_extra=_extra) eq 0)then \$
> return, 0
> ;file
> self->RegisterOperation, 'Exit', 'IDLitopFileExit', \$
> IDENTIFIER='File/Exit', /SEPARATOR
>
> ;button
> self->RegisterManipulator, 'Arrow', 'IDLitManipArrow', \$
> ICON='arrow', /DEFAULT, IDENTIFIER="ARROW", \$
> DESCRIPTION='Click to select items, click and drag to select
multiple items'
>
> return, 1
>
> END
>
> That gets you onto the first step...
>
> Chris.
