Subject: Re: Managing several object instances
Posted by David Fanning on Mon, 17 Sep 2012 22:17:41 GMT
View Forum Message <> Reply to Message

Helder writes:

> However, I'm not coming around a problem. How can I efficiently keep track of the generated object instances. Suppose the user generates 5 line profiles, and destroys the third and then generates a new one and so on and so forth.
> What is the most efficient way of keeping track of the existing/running object instances so that I can efficiently pass around my token?
> That means that I need to be able to add and delete object instances in a record and at any time be able to know which ones are working and active...

Why would you want to keep track of them?

Just write a program that does what you want it to do
(resize, save, etc.) and send it forth from your other
program. When the user is finished with it, he or she
will kill it. What do you care?

Make sure you call the program with a Group_Leader, which
will be the top-level base of your GUI. Then, when your
GUI dies, all the programs that were spawned from it
(however many are still left!) will be destroyed, too.

Cheers,

David

P.S. You could use cgWindow to display your image profiles,
for example, then you already have all the resizing and saving
built right in. You really don't have anything at all to do,
except fill up the windows with the right commands.

--
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: Managing several object instances
Posted by Helder Marchetto on Tue, 18 Sep 2012 05:28:16 GMT
View Forum Message <> Reply to Message

On Tuesday, September 18, 2012 12:17:42 AM UTC+2, David Fanning wrote:
> Helder writes:

>
>
>
>> However, I'm not coming around a problem. How can I efficiently keep track of the generated object instances. Suppose the user generates 5 line profiles, and destroys the third and then generates a new one and so on and so forth.
>
>> What is the most efficient way of keeping track of the existing/running object instances so that I can efficiently pass around my token?
>
>> That means that I need to be able to add and delete object instances in a record and at any time be able to know which ones are working and active...
>
>
>
> Why would you want to keep track of them?
>
>
>
> Just write a program that does what you want it to do
>
> (resize, save, etc.) and send it forth from your other
>
> program. When the user is finished with it, he or she
>
> will kill it. What do you care?
>
>
>
> Make sure you call the program with a Group_Leader, which
>
> will be the top-level base of your GUI. Then, when your
>
> GUI dies, all the programs that were spawned from it
>
> (however many are still left!) will be destroyed, too.
>
>
>
> Cheers,
>
>
>
> David
>
>
>
> P.S. You could use cgWindow to display your image profiles,

>
> for example, then you already have all the resizing and saving
>
> built right in. You really don't have anything at all to do,
>
> except fill up the windows with the right commands.
>
>
>
> --
>
> 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.")

Thanks for the answer David,
however I'm still puzzled by the fact that when something changes in the main program (the image), the object instances have to be modified. If I don't know which instances are running I don't know which window I should tell something.

There is probably an easy solution and it's there right in front of me, but I seem to be running around it.

Cheers,
Helder

PS: Using cgWindow is good, but I would like to add some extra functionality to the plot so that the user can change the line thickness, style, color,... Since I like reinventing wheels, I've managed the resizable graphics and saving on my own with quite a few limitations, but it does the job.

Subject: Re: Managing several object instances
Posted by Paul Van Delst[1] on Wed, 19 Sep 2012 14:12:11 GMT
View Forum Message <> Reply to Message

Hello,

Without addressing the "why are you doing this", what about:

1) using a List, or a Hash, object to store your objects? You can use
their Add (for Lists)
   and Remove methods to manage your own object collection.
2) create your own object handling class that inherits the IDL_Container

class - which
 also has its own Add and Remove methods

I was also going to suggest extending (2) a bit an generate your own
linked list, but I don't
think your usage requires a linked list.

I would start with (1) since lists are a subclass of IDL_Container already:

```
IDL> x=list()
IDL> help, x, /object
** Object class LIST, 2 direct superclasses, 1 known method
   Superclasses:
      COLLECTION <Direct>
      IDL_OBJECT
      IDL_CONTAINER <Direct>
   Known Function Methods:
      LIST::INIT
```

cheers,

paulv

Subject: Re: Managing several object instances
Posted by Helder Marchetto on Wed, 19 Sep 2012 15:45:24 GMT

View Forum Message <> Reply to Message

Hi

On Wednesday, September 19, 2012 4:11:58 PM UTC+2, Paul van Delst wrote:
> Hello,
>
>
>
> Without addressing the "why are you doing this", what about:
>
>
>
> 1) using a List, or a Hash, object to store your objects? You can use
>
> their Add (for Lists)
>
>   and Remove methods to manage your own object collection.
>
> 2) create your own object handling class that inherits the IDL_Container
>

> class - which
>
>   also has its own Add and Remove methods
>
>
>
> I was also going to suggest extending (2) a bit an generate your own
>
> linked list, but I don't
>
> think your usage requires a linked list.
>
>
>
> I would start with (1) since lists are a subclass of IDL_Container already:
>
>
>
> IDL> x=list()
>
> IDL> help, x, /object
>
> ** Object class LIST, 2 direct superclasses, 1 known method
>
>   Superclasses:
>
>   COLLECTION <Direct>
>
>   IDL_OBJECT
>
>   IDL_CONTAINER <Direct>
>
>   Known Function Methods:
>
>   LIST::INIT
>
>
>
>
>
> cheers,
>
>
>
> paulv

Dear Paul and David,
thank you very much for your suggestions. Thank you for sharing your thoughts on this.

Since your experience with writing software is beyond reach for me, I'll try to pick up on your suggestions to avoid making a list.
However every time I think about this, I still cannot figure out how to manage the whole together.
Suppose I have an "image manager" that loads an image from a stack. Image n is selected an on this image there are 5 lines. The users selects line 3 and opens a plot profile window. This means I initiate an instance of the object that displays the data. Then another instance is initiated with line 2 and a new profile window appears. Now a new image in the image manager is selected and line 1 is deleted. I need to update one or all of the plot profile window/objects. Don't I need to know which ones have been initiated and are still running and to which line each object is/was associated ?
This is the only reason why I want to use a list where I store the object and associate the line that is connected to this object. I see no other way out of this.

If there is a "proper" way to do this avoiding lists, I'm would be happy to learn that because this sort of problem I have quite often...

Cheers,
Helder

Subject: Re: Managing several object instances
Posted by DavidF[1] on Wed, 19 Sep 2012 16:29:21 GMT

View Forum Message <> Reply to Message

Helder writes:

> Since your experience with writing software is beyond reach for me, I'll try to pick up on your suggestions to avoid making a list.
>
> However every time I think about this, I still cannot figure out how to manage the whole together.
>
> Suppose I have an "image manager" that loads an image from a stack. Image n is selected an on this image there are 5 lines. The users selects line 3 and opens a plot profile window. This means I initiate an instance of the object that displays the data. Then another instance is initiated with line 2 and a new profile window appears. Now a new image in the image manager is selected and line 1 is deleted. I need to update one or all of the plot profile window/objects. Don't I need to know which ones have been initiated and are still running and to which line each object is/was associated ?
>
> This is the only reason why I want to use a list where I store the object and associate the line that is connected to this object. I see no other way out of this.
>
> If there is a "proper" way to do this avoiding lists, I'm would be happy to learn that because this sort of problem I have quite often...

I don't see any way out of it, either, given your stated requirements. I'm just suggesting a couple of hours spent thinking about how to avoid those requirements might be time well spent. :-)

Sometimes there is no way out of it. Paul's suggestion to use a LIST is a good one. Before IDL 8, I have used either a Linked List (from the Coyote Library) or my own modified IDL_Container object (in the Catalyst Library). Our modfied container object had the option to be an indexed container or just a container. An indexed container is handy for keeping track of things like you describe, if you can find some way to keep it up to date as objects go into and out of service. (An image object could contain an index container with five spaces for the five allowed profile objects, for example. That way, it could keep track of its own profiles.)

But, if it were me, when a new image was selected, I would destroy ALL of the profile objects wantonly, so that I could start fresh. Figuring out how to update displays that may or may not be on the display sounds like a nightmare to me. :-)

Cheers,

David