
Subject: Re: object reference destroy

Posted by [David Fanning](#) on Mon, 22 Jun 2009 12:27:56 GMT

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

crane writes:

> I have a container to hold a object, and I need to use this object
> often. In order to avoid memory leakage, i have to destroy the object
> reference after use. But once i destroy the object reference, the
> object in the container is also destroy, because they are pointing to
> the same object. For example:
> ...
> I like to use container to hold object, it is convenience to play with
> objects. can somebody help me through this problem or have another
> idea for alternative.

You might want to have a look at the CatContainer object in my Catalyst Library code. It reference counts objects that are added to it by adding a "parent" to the added object's "people interested in me" list. When the container is destroyed it takes its name off the "people interested in me" list and the added object is only destroyed if the container sees there is no more interest in the object. That is, the object is destroyed if the container was the last "person" interested in the object.

Provisions are made in the object to **never** destroy the object's added to it, if that is what you would prefer, too.

http://www.dfanning.com/programs/catalyst/source/core/catcontainer__define.pro

Your objects would simply inherit CatContainer to have this functionality.

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

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

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

Subject: Re: object reference destroy

Posted by [greg.addr](#) on Mon, 22 Jun 2009 12:40:59 GMT

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

```
;put a cake into the container
container = obj_new('IDLtin')
cake = obj_new('IDLcake')
container->add, cake
```

```
;get the cake out for the first time
objref_1= container->get(/all)
```

```
....
```

```
;eat the cake for the first time. (hunger reason)
obj_eat, objref_1
```

```
;need to get the cake out from the container second time
objref_2= container->get(/all)
```

```
;but objref_2 is not a cake, because i ate it in the previous
section.
```

```
IDL> print, obj_valid(objref_2)
0
```

On Jun 22, 11:53 am, crane <dos...@163.com> wrote:

```
> hi, people
>
> I have a container to hold a object, and I need to use this object
> often. In order to avoid memory leakage, i have to destroy the object
> reference after use. But once i destroy the object reference, the
> object in the container is also destroy, because they are pointing to
> the same object. For example:
>
> ; put a object into the container
> container = obj_new('idl_container')
> obj = obj_new('myclass')
> container->add, obj
>
> ; get the object for the first time
> objref _1= container->get(/all)
> ....
> ; destroy the object reference for the first time. (memory reason)
> obj_destroy, objref_1
>
> ; need to get the object from the container second time
> objref_2= container->get(/all)
> ; but objref_2 is not a valid object anymore, because i destroy the
> object in the previous section.
> IDL> print, obj_valid(objref_2)
```

> 0
>
> I like to use container to hold object, it is convenience to play with
> objects. can somebody help me through this problem or have another
> idea for alternative. thanks.

Subject: Re: object reference destroy

Posted by [Doug Edmundson](#) on Mon, 22 Jun 2009 16:30:59 GMT

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

Hi crane,

The get() method returns a reference to the stored object - not some kind of copy of the original. So when you delete the gotten object, the original is destroyed.

You can eat as much of your cake as you want, just don't throw it away until you really don't want the cake any more. ;-)

Hope this helps,
Doug

crane wrote:

> hi, people
>
> I have a container to hold a object, and I need to use this object
> often. In order to avoid memory leakage, i have to destroy the object
> reference after use. But once i destroy the object reference, the
> object in the container is also destroy, because they are pointing to
> the same object. For example:
>
> ; put a object into the container
> container = obj_new('idl_container')
> obj = obj_new('myclass')
> container->add, obj
>
> ; get the object for the first time
> objref _1= container->get(/all)
>
> ; destroy the object reference for the first time. (memory reason)
> obj_destroy, objref_1
>
> ; need to get the object from the container second time
> objref_2= container->get(/all)
> ; but objref_2 is not a valid object anymore, because i destroy the
> object in the previous section.
> IDL> print, obj_valid(objref_2)

> 0
>
>
> I like to use container to hold object, it is convenience to play with
> objects. can somebody help me through this problem or have another
> idea for alternative. thanks.
>
>
>
>
>
>
>
>
>
>

Subject: Re: object reference destroy
Posted by [BlackMage](#) on Wed, 24 Jun 2009 18:33:39 GMT
[View Forum Message](#) <> [Reply to Message](#)

An alternative is to implement reference counting. I have implemented this for my own code and found it very effective, all my classes have the reference counting class as the root of their inheritance. When the reference-counting object is created or another object tells it that it 'owns' a reference to it, the count increases. Once other objects no longer need the owning reference, they tell the reference-counting object and the count decreases. Once the count hits zero, the reference-counting object destroys itself. If the reference-counting object is destroyed with a non-zero count, it complains and this assists debugging.

This approach has been very effective for me in object management as I am no longer trying to work out exactly what part of my code holds the final reference to any given object so I can destroy it when, and only when, it is no longer needed. It's especially useful if you have multiple collections of the same object references e.g. search trees or linked lists.

Subject: Re: object reference destroy
Posted by [David Fanning](#) on Wed, 24 Jun 2009 18:42:35 GMT
[View Forum Message](#) <> [Reply to Message](#)

BlackMage writes:

> An alternative is to implement reference counting

Humm. Thought I mentioned this. ;-)

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

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

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