
Subject: idl_conatiner::get and position
Posted by [btt](#) on Thu, 24 Feb 2005 21:58:57 GMT
[View Forum Message](#) <> [Reply to Message](#)

Oh dear,

When GETting objects in a container the POSITION keyword specifies the location of the objects needed. But they aren't returned in that order. They are returned in the order they are stored (first in - first out).

Is there a way around this other than writing a method override?

Here's an example.

```
x = obj_new('idl_container')
for i = 0, 5 do x->Add, obj_new('idlgrmodel')
```

```
arr = x->Get(/all)
for i = 0, 5 do print, arr[i]
```

```
<ObjHeapVar847(IDLGRMODEL)> <<<< note the order
<ObjHeapVar849(IDLGRMODEL)>
<ObjHeapVar851(IDLGRMODEL)>
<ObjHeapVar853(IDLGRMODEL)>
<ObjHeapVar855(IDLGRMODEL)>
<ObjHeapVar857(IDLGRMODEL)>
```

```
y = x->get(position = [3,2,1]) <<<< note the position order
for i = 0, 2 do print, y[i]
```

```
<ObjHeapVar849(IDLGRMODEL)> <<<< note the order
<ObjHeapVar851(IDLGRMODEL)>
<ObjHeapVar853(IDLGRMODEL)>
```

Any ideas?

Thanks,

Ben

{ ppc darwin unix Mac OS X 6.1.1 Oct 11 2004 32 32}

Subject: Re: idl_conatiner::get and position
Posted by [David Fanning](#) on Mon, 28 Feb 2005 15:23:46 GMT
[View Forum Message](#) <> [Reply to Message](#)

Ben Tupper writes:

> Interesting. I haven't had the need to preserve the space occupied by a removed
> object.
>
> Here's how I think I'll handle to first-in first-out ordering (or at least
> something like this.)

Well, did you try that? :-)

At first glance, it looks like you are going to have
the same problem. I think your loop will have to go
like this:

```
FOR j=(NP-1),0,-1 DO
```

Otherwise, your indexing will get screwed up again, won't it?
I don't have time to check this morning, alas. :-)

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

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

Subject: Re: idl_conatiner::get and position
Posted by [btt](#) on Mon, 28 Feb 2005 16:45:54 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning wrote:

> Ben Tupper writes:
>
>
>> Interesting. I haven't had the need to preserve the space occupied by a removed
>> object.
>>
>> Here's how I think I'll handle to first-in first-out ordering (or at least
>> something like this.)
>
>
> Well, did you try that? :-)
>
> At first glance, it looks like you are going to have
> the same problem. I think your loop will have to go
> like this:

```

>
>   FOR j=(NP-1),0,-1 DO
>
> Otherwise, your indexing will get screwed up again, won't it?
> I don't have time to check this morning, alas. :-)
>

```

I always get that sinking feeling when you ask if I'm sure. But in this case I'm fully 3.93% confident that I am ok on this. Here's an example. Compile it, then run the TEST_MYCONTAINER. Here's what I get for output...

```

IDL> test_myContainer
FIFO order
<ObjHeapVar779(IDLGRFONT)> < this shows the First-In First-out order [0,2,3]
<ObjHeapVar783(IDLGRFONT)>
<ObjHeapVar785(IDLGRFONT)>

```

```

NoSort order
<ObjHeapVar785(IDLGRFONT)> < this shows the desired order [3,0,2]
<ObjHeapVar779(IDLGRFONT)>
<ObjHeapVar783(IDLGRFONT)>

```

```

;*****BEGIN CODE
PRO Test_MyContainer

```

```

myCon = OBJ_NEW('myContainer')

```

```

For i = 0, 5 do myCon->Add, OBJ_NEW('IDLgrFont')

```

```

fifo = myCon->Get(position = [3,0,2])
Print, 'FIFO order'
For i = 0, 2 do print, fifo[i]
Print, "
myOrder = myCon->Get(position = [3,0,2], /NOSORT)
Print, 'NoSort order'
For i = 0, 2 do print, myOrder[i]

```

```

OBJ_DESTROY, myCon
END

```

```

FUNCTION myContainer::Get, $
  COUNT = count, $
  NOSORT = nosort, $
  POSITION = position, $

```

```

    _EXTRA = extra

nP = n_elements(position)
If keyword_Set(NoSort) AND (nP GT 1) Then Begin
    arr = objarr(nP)
    For i = 0, nP-1 Do $
        arr[i] = self->Get(position = position[i], _EXTRA = extra)
    count = nP
EndIf Else Begin
    arr = $
    self->IDL_CONTAINER::Get(position = position, $
        COUNT = count, _EXTRA = extra)
EndElse

Return, arr
END

```

```

FUNCTION myContainer::Init
Return, self->IDL_CONTAINER::Init()
END

```

```

PRO myContainer::Cleanup
Self->IDL_CONTAINER::Cleanup
END

```

```

PRO myContainer__Define, class
class = {myContainer, INHERITS IDL_CONTAINER}
END

```

```

;*****FINI CODE

```

Subject: Re: idl_conatiner::get and position
 Posted by [David Fanning](#) on Mon, 28 Feb 2005 17:16:55 GMT
[View Forum Message](#) <> [Reply to Message](#)

Ben Tupper writes:

> I always get that sinking feeling when you ask if I'm sure. But in this case I'm
 > fully 3.93% confident that I am ok on this. Here's an example. Compile it,
 > then run the TEST_MYCONTAINER. Here's what I get for output...

Oh, right, sorry. I thought for some reason we were doing REMOVE
 and not GET. Yes, GET will not change the order of the index,
 since it is not removing anything from the container.

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

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