
Subject: Re: idl_container::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
```
