Subject: Re: idl_conatiner::get and position
Posted by David Fanning on Fri, 25 Feb 2005 00:01:44 GMT
View Forum Message <> Reply to Message

Ben Tupper writes:

> 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?

We have run into this problem, too. Our sub-classed container
can either be normal or "indexed", in which case when you
remove something from a container it is replaced by a null
object, so that the position of the other objects in the container
is preserved.

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 15:16:53 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
> Ben Tupper writes:
>
>
>> 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?
>
>
> We have run into this problem, too. Our sub-classed container
> can either be normal or "indexed", in which case when you
> remove something from a container it is replaced by a null
> object, so that the position of the other objects in the container
> is preserved.

>

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.)

```
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
```

Thanks,
Ben
