
Subject: Re: idl_container::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
