
Subject: Re: array of objects?

Posted by [Ted Cary](#) on Tue, 12 Feb 2002 14:37:46 GMT

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

I can't check this now, but my guess is that you replicated the object reference when you created your array--you have 100 references to the same object, not 100 different instances of the object. In other words, Obj_New was only called one time, and the reference it returned was put into every element of the array. Check your heap (Help, /Heap) --are there 100 objects, or just one? All you did in each iteration of your SetProperty loop was set the object's self.integer_value to the loop counter, so the highest count is returned by GetProperty.

HTH

Subject: Re: array of objects?

Posted by [Pavel A. Romashkin](#) on Tue, 12 Feb 2002 17:22:20 GMT

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

Ted is correct. For creating an array of objects, you need to use OBJARR function, then loop through the array and populate each element using OBJ_NEW. What you have created in your code is 100 references to the same object.

We talked before about the advantages of allowing definitions of object arrays the same way as normal arrays are made, but the consensus was that it is unsafe :(because, say, one could populate an object array with different object classes. Personally, I think it is not so and I'd leave that decision to the programmer, but heap variable arrays are slower a bit anyway, so none of the fast vector features of IDL would've worked on them anyway.

HTH,
Pavel

Thomas Bielser wrote:

```
>
> ...but IDL comes up with:
>
> IDL> print, a[0] -> getproperty( /integer_value )
>    99
> IDL> print, a[50] -> getproperty( /integer_value )
>    99
> IDL> print, a[99] -> getproperty( /integer_value )
>    99
>
> Is this a feature or a bug? Many thanks in advance!
> Thomas
```

Subject: Re: array of objects?

Posted by [thomas.bielser](#) on Wed, 13 Feb 2002 07:19:57 GMT

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

Thanks to Ted and Pavel: Your were absolutely right!

After creating the array with 100 elements (each with a null pointer):

```
; array for holding 100 elements of the type 'own_class'  
a = OBJARR( 100 )
```

I added the following statement in the loop, which runs over all the elements:

```
; label each array-element with it's number  
for i=0, 99 do begin  
  
; create a new object and put the point to it via the variable "a"  
a[ i ] = OBJNEW( 'own_class'  
  
; call the method setproperty  
a[ i ] -> setproperty, integer_value = i  
  
endfor
```

Afterwards I had the expected output as mentioned in my former posting.

Cheers,
Tom
