## Subject: Re: Modifying Arrays and Structures in HASH's (hint: you can't)
## Posted by m_schellens on Mon, 29 Jul 2013 08:46:36 GMT

View Forum Message <> Reply to Message

Am Sonntag, 28. Juli 2013 21:32:25 UTC+2 schrieb fawltyl...@gmail.com:

> On Sunday, July 28, 2013 6:13:18 AM UTC+2, bobnn...@gmail.com wrote:
>
>
>
>> I will say that the inability of accessing directly into hashes and lists (by reference, not copy) is so disappointing that I am moving away from IDL. So if you found a way around this it would be very nice.
>
>
>
> LIST is a pointer array in disguise. You can get a copy of this array and manipulate list elements directly:
>
>
>

```
>  IDL> l=list(1,2,3)
>
>  IDL> a=l.idl_container::get(/all)
>
>  IDL> print, l
>
>        1
>
>        2
>
>        3
>
>  IDL> *a[1]=123
>
>  IDL> print, l
>
>        1
>
>      123
>
>        3
```

>
>
>
> regards,
>
> Lajos

As of IDL 8.0, this is not correct. An IDL LIST is really a sinlge linked list made up of (PTR) heap variable nodes (IDL_CONTAINER_NODE). The IDL_CONTAINER::GET function creates then the array.

But your method works, as the (copied) pointers access the same heap variables. This is also the core of the mechanism I suggested for _OVERLOADBRACKETSLEFTSIDE.

Also note, that at least with HEAP the IDL_CONTAINER::GET functionality cannot work anymore (as you cannot pick the right element).

And it is of course as well not efficient, to convert the complete container to a pointer array in order to left-access one element.
And the call to GET is almost as ugly as copying out one element, left-accessing it and copying it back.

Regards,
Marc