Subject: Re: Arrays of Structures
Posted by Brian Larsen on Thu, 08 Feb 2007 17:41:48 GMT
View Forum Message <> Reply to Message

Mick,

you just need to dig a little deeper with help for the answer.

This is what you did and it is true this is an array with 50 elements

```
IDL>  HELP, structs.a
<Expression>    INT       = Array[50]
```

but if you look at structs by itself not structs.a, it is the array,
NOT structs.a

```
IDL>  HELP, structs
STRUCTS         STRUCT    = -> <Anonymous> Array[50]
```

So what is inside the structure?

Here a is inside the struct and it is an int

```
IDL>  HELP, structs, /str
** Structure <84bd40c>, 1 tags, length=2, data length=2, refs=1:
   A               INT              1
```

Meaning that structs.a[23] doesn't make sense because structs.a is an int. While structs[23].a does make sense because structs is an array.

This is just one of those lessons that take a while to get a hold of.

Make any sense?

Brian

 -- ---------
Brian A. Larsen
Dept. of Physics
Space Science and Engineering Lab (SSEL)
Montana State University - Bozeman
Bozeman, MT 59717

On Feb 8, 10:03 am, "Mick Brooks" <mick.bro...@gmail.com> wrote:
> Hi,
>

> Can anyone help my understanding of what happens when I apply a tag-name to an array of structures? All of my previous questions have been answered by searching on eitherwww.dfanning.comor this newsgroup, but this one has me stumped.
>
> Take an array of (very boring, anonymous) structures:
>
> ```
> IDL> structs = REPLICATE({a:1}, 50)
> ```
>
> If I look-up a tag-name on this array, I see this:
>
> ```
> IDL> HELP, structs.a
> <Expression>    INT       = Array[50]
> ```
>
> That looks like wonderful magic to me (I don't have much experience with array-based languages) - IDL knew to apply the tag-name to each structure in turn, and return me an array of the values - in this case an array of INTs. Now what if I want the 3rd element of this array? Let's try:
>
> ```
> IDL> HELP, structs.a[2]
> % Subscript range values of the form low:high must be >= 0, < size,
> with low
>   <= high: <No name>.
> % Execution halted at: $MAIN$
> ```
>
> Not so good. I know of two workarounds for this. Either take the 3rd element of the array of structures before looking up the tag-name (I understand why this one works):
>
> ```
> IDL> HELP, structs[2].a
> <Expression>    INT       =        1
> ```
>
> Or, put some extra parentheses in (I've no idea why this one fixes it):
>
> ```
> IDL> HELP, (structs.a)[2]
> <Expression>    INT       =        1
> ```
>
> Why does my first attempt fails, and why do the parentheses help?
>
> For more confusion, look what happens if I try and lookup the 2nd, 4th and 28th element all at once:
>
> ```
> IDL> indices = [1, 3, 27]
> IDL> HELP, structs.a[indices]
> <Expression>    INT       = Array[3, 50]
> ```
>

> Where did that extra dimension come from? What is the type of
> structs.a? It seems that if I don't put parentheses around it, some of
> the magic leaks out...
>
> Cheers,
>
> Mick Brooks