Subject: Arrays of Structures
Posted by Mick Brooks on Thu, 08 Feb 2007 17:03:00 GMT
View Forum Message <> Reply to Message

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 either www.dfanning.com or 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