Subject: Re: Arrays of Structures
Posted by Mick Brooks on Thu, 08 Feb 2007 19:27:48 GMT
View Forum Message <> Reply to Message

On Feb 8, 7:09 pm, "R.G. Stockwell" <n...@email.please> wrote:

>> ... but this is where you lose me. Doesn't the the first line that you
>> showed say that structs.a is an array of 50 ints?
>
> Not exactly. The key point is that structs.a is an EXPRESSION.
> Also, "struct" is an array of structures.

Yes, this struck me as I drove home from work. I don't have IDL here,
but will be trying
IDL> INDGEN(50)[23]
first thing in the morning

> You cannot do
> IDL> EXPRESSION(10)
> , you must first cast the expression
> IDL> (EXPRESSION)[10]
>
> It is not an operator precedence thing.

Yes, I'd looked at the precedence table and made the same conclusion
as you and Paulv.

All that's left is to explain why we get the extra leading dimension
when subscripting an expression-which-evaluates-to-an-array-of-ints
thing (i.e. structs.a) with multiple indices (or [*]).
This was the actual question I was asked by a user of the module I'd
written - he was seeing these extra leading dimensions, and queried
whether my structure was somehow broken. I couldn't answer, and so
came up with the simpler case we're discussing.

Thanks for all your help,

Mick