
Subject: Re: Language Documentation;
Posted by [steinhh](#) on Fri, 08 Jul 1994 16:55:05 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <2vjsic\$9p5@ncar.ucar.edu>, caron@acd.ucar.edu (John Caron) writes:
> As a general comment to RSI, your manuals seem written for the casual,
> scientist-type (i.e. non-programmer). Sort of a "dont confuse them" attitude.
> Your language description is woefully vague to the eyes (ears?) of this
> programmer. How about a "programmer's description" of the language?
>
> Comments?
>

Quite agree!

I was very bewildered one day when someone commented that my use of "not 0" as a return value could cause problems, since some Fortran users would write their logical tests as "if i eq 1 then..", meaning "if i then". With an explanation of the rules, this could be avoided.

Along the same lines, why shouldn't it be possible to have an anonymous structure as a member of another structure?

And could I please have *pointers* (reference data type)? In Fortran-style programs (ahem, read scientist's programs) it's quite useless, but when working with e.g. complex widget applications, sharing large amounts of data, I've had to implement a "pointer" data type by using SET/GET_UVALUE and the /NO_COPY keyword! It's actually quite effective (compared to the alternatives), but not really easy-to-use!

And the manual could have warned about the fact that the use of the NO_COPY keyword actually makes the variable undefined, when setting a UVALUE..

Well, that's enough for today..

Stein Vidar Haugan

Subject: Re: Language Documentation;
Posted by [rmm](#) on Wed, 13 Jul 1994 15:32:16 GMT
[View Forum Message](#) <> [Reply to Message](#)

In IDL v3.6 you may have anonymous structures as members of anonymous structures. You can also use the new "HANDLES" routines to implement pointer type operations. I'm fairly certain that these are using the uvalue of unrealized widget_bases. The effect is very similar to a true pointer, except that a pointer is typically 4 bytes, whereas these handles are probably on the order of 200 bytes. They do work well, however.

Robert M. Moss, Ph.D.
Texaco Inc.
rmoss@texaco.com

#include "/std/disclaimer"
