Subject: Re: Functions and arrays
Posted by rigby on Wed, 04 Dec 1996 08:00:00 GMT
View Forum Message <> Reply to Message

In article 6ll@post.gsfc.nasa.gov, thompson@orpheus.nascom.nasa.gov (William Thompson) writes:
>
> I've just run into a serious limitation in the way IDL handles functions and
> arrays. I've always been aware that there was an ambiguity in the IDL syntax,
> ...
> It is my opinion that IDL should be tightened in its handling of arrays and
> functions. If a procedure contains a variable with a given name, then it
> should be unambiguous that one is referencing that variable, and not some
> function which happens to share the same name.
> ...
> William Thompson

Please complain directly to RSI about this! We ran into this about a year ago, and were unable to get past the first level support folks telling us that this was an inevitable consequence of IDL's weak typing.

We even gave them an example of how their own routines could be broken. The routine poly.pro in $IDL_DIR/lib happens to use the name C for one of its formal arguments:

```
        FUNCTION POLY,X,C
```

If a user should compile a function C before POLY is compiled, he/she will get the wrong answer from POLY. Silently.

```
IDL> .run
- function c, arg
- return, arg-1000
- end
 Compiled module: C.
IDL> print, poly(1,[1,1])
 Compiled module: POLY.
     -1999              (should be 2).
```

There's no reasonable way to defend yourself against this either.
Are you supposed to examine the formal argument list of every routine you use??

Wayne Rigby
GE Corporate Research and Development
rigby@crd.ge.com