
Subject: Re: check for duplicate routine names?

Posted by [Liam E. Gumley](#) on Fri, 19 Jan 2001 16:52:51 GMT

[View Forum Message](#) <> [Reply to Message](#)

Craig Markwardt wrote:

> Another thing that would be nice is warnings about ambiguous uses of
> variables that might shadow a function in the library. Note this
> code:
>
> max = max(x, min=min)
>
> The next time you use min(x), will you get the function or the
> variable named min?
>
> You might say, "oh, I used square brackets, this is not a problem,"
> but since round parenthesis are still allowed (unless you disable
> them), I believe this can still be a problem. This has bitten me a
> *few* times, so I'm not saying it's a huge deal. It is a compile time
> issue since peoples' function libraries are different.

I believe a function takes precedence over an identically named array subscripted with parentheses, e.g.

```
IDL> min = indgen(10)
IDL> a = indgen(5)
IDL> print, min(a)
0
```

Using square brackets eliminates the ambiguity between the function name and the variable name:

```
IDL> print, min[a]
0    1    2    3    4
```

If you want to check whether a built-in or library function already exists with a certain name (such as 'MIN'), first check the online help:

```
IDL> ? min
```

If it's not found there, perhaps there is a library function named min.pro, and you could try compiling it:

```
IDL> .compile min
```

The only remaining possibility (I think) is that a function or procedure with the same name is buried inside some other procedure or function yet to be compiled. In any case, using square brackets for array subscripting helps minimize name-space problems.

Cheers,
Liam.
<http://cimss.ssec.wisc.edu/~gumley>
