

---

Subject: Re: arrays vs. functions conflicts

Posted by [Paul Van Delst\[1\]](#) on Fri, 05 Mar 2004 14:54:58 GMT

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

---

Paolo Grigis wrote:

```
>
> David Fanning wrote:
>> Paolo Grigis writes:
>>
>>
>>> Thus my problem:
>>> To resolve the conflict {that is, limits.pro being already
>>> compiled and procedure.pro refusing to compile because it has
>>> an old-fashioned statement like var=limits(1:2) instead of
>>> var=limits[1:2]} I'm thinking of automatically compiling
>>> the (hopefully) few troubling routines like procedure.pro
>>> at startup using the resolve_routine() statement.
>>> (BTW, why is the IDL compiler (5.5) not smart enough to
>>> understand that function(1:2) is an array? ":" is never allowed
>>> in function calls, after all.)
>>>
>>> But before going on, I just wanted to know if there is an
>>> easier way out of this that I have overlooked.
>>
>>
>> IDL itself could care less about this issue. So if
>> you are having problems with it then *you* must
>> care about it. Does your procedure.pro have a compiler
>> option that forces strict arrays? Then take it out.
>> Problem solved. :-)
>>
>> Cheers,
>>
>> David
>
> No, the routine compiles just fine if there aren't
> any previously compiled functions called "limits", but
> *fails* to compile if this is the case, because then IDL
> thinks it is a function instead of an array.
> Hence I was thinking of compiling the routine at
> the idl start: if I do that then I don't have any problems
> at all.
```

Hello Paolo,

Is there a COMPILE STRICTARR directive anywhere in your code or in any startup scripts?  
This is the most obvious source of weirdness between [] and (). But.....

Hang on a minute.... you say you have a limits.pro that compiles. Thus "limits" \_is\_ a function, right? The you have a statement like var=limits(1:2) where "limits" is now an array? Well, which do you want "limits" to be...a function or an array?

Confusedly yours,

paulv

--

Paul van Delst  
CIMSS @ NOAA/NCEP/EMC

---