Subject: Re: arrays vs. functions conflicts
Posted by Paolo Grigis on Fri, 05 Mar 2004 15:17:14 GMT
View Forum Message <> Reply to Message

Paul van Delst wrote:
> 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 ouy 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 you 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
>

Dear Paul, that's the problem: in the procedure, limits is an
array. No problem with that, it should be.

But sometimes it happens that I run *another* program before,
which compile a function called limits. If then I compile the
procedure, the compiler thinks limits is a function and all
hell break loose.

The problem is that in general you don't have control on what
other people define as a function, and so this problem does
happen. The only way out I see, is to exit idl, and compile
the procedure before the function.

Cheers,
Paolo

--

Paolo Grigis
ETHZ - Institute of Astronomy email: pgrigis@astro.phys.ethz.ch
Scheuchzerstrasse 7
ETH Zentrum phone: ++41 1 632 42 20
8092 Zurich fax : ++41 1 632 12 05
Switzerland http://www.astro.phys.ethz.ch/