Subject: Re: dereference with wildcard
Posted by Carsten Lechte on Tue, 09 Feb 2010 17:05:08 GMT
View Forum Message <> Reply to Message

PimK wrote:
> print, mean((*(*(*pState).navtrack)[*]).lat)
> % Expression must be a scalar in this context: <POINTER   Array[2]>.

This is because you can only dereference scalar pointers. This is a fundamental limitation of IDL's arrays: all array elements have to be of the same scalar type. You can have arrays of pointers to anything, since the array is made up of things of the same type (i.e. pointers). If you dereferenced the whole array, you would have an array of "anything", which is not allowed. IDL refuses to evaluate the expression, even if all the pointers pointed to a scalar integer, and the resulting array would be legal in IDL.

chl

Subject: Re: dereference with wildcard
Posted by PimK on Wed, 10 Feb 2010 20:16:32 GMT
View Forum Message <> Reply to Message

On Feb 9, 1:05 pm, Carsten Lechte <c...@toppoint.de> wrote:
> PimK wrote:
>> print, mean((*(*(*pState).navtrack)[*]).lat)
>> % Expression must be a scalar in this context: <POINTER   Array[2]>.
>
> This is because you can only dereference scalar pointers. This is a fundamental
> limitation of IDL's arrays: all array elements have to be of the same scalar
> type. You can have arrays of pointers to anything, since the array is made up
> of things of the same type (i.e. pointers). If you dereferenced the whole array,
> you would have an array of "anything", which is not allowed. IDL refuses to
> evaluate the expression, even if all the pointers pointed to a scalar integer,
> and the resulting array would be legal in IDL.
>
> chl

Thanks for the feedback! I'll use loop then instead.
Cheers, P.