Subject: Re: behavior of arrays
Posted by Jack Saba on Thu, 20 May 1999 07:00:00 GMT
View Forum Message <> Reply to Message

But more often than not, I WANT the extra dimension to be lost,
or at least I want IDL to be willing to ignore it where appropriate.
Consider this unrealistic example that nevertheless illustrates a
problem that occurs all too often in IDL:

IDL> x=findgen(100)
IDL> ijk=where(x eq 10)
IDL> for i=ijk,99 do print, i
% Expression must be a scalar in this context: I.
% Execution halted at: $MAIN$

I could have written i=ijk[0],99, or i=REFORM(ijk),99 to avoid the
error. But it shouldn't be necessary -- this should be handled
transparently.

"R.Bauer" wrote:
>
> I have some problems to understand the logic which may be behind the
> handling of an array
> like defined.
>
> Why is the result help, b.d only [10] and not [10,1] ?
>
> This means I will lose the information that's it is / was a 2-dim
> dataset.
>
> R.Bauer
>
> d=reform(findgen(10),10,1)
> help,d
> ;D   FLOAT   = Array[10, 1]
>
> b=create_struct('d',d)
> help,b,/str
>
> ;** Structure <1348428>, 1 tags, length=40, refs=1:
> ;   D   FLOAT   Array[10, 1]
>
> help,b.d
> ; <Expression>   FLOAT   = Array[10]
>
> c=b.d
> help,c

> ; C FLOAT = Array[10]