Subject: Re: Reading files with unknown amount of dat
Posted by zawodny on Fri, 05 Nov 1993 12:51:07 GMT
View Forum Message <> Reply to Message

In article <2bd2qgINNkpn@maz4.sma.ch> oet@sma.ch writes:
> In Perl exists a function 'push' to dynamically grow up arrays. Some times
> ago I wrote a similiar function in IDL. The function requires the function
> datatype.pro from the JHUAPL-IDL-Library.
> Both functions are included below.
>
> --Thomas

Stuff deleted

>
> type = datatype(oldarr,3)
> newdim=n_elements(oldarr)+n_elements(newvals)
> old_last=n_elements(oldarr)-1
> new_last=newdim-1
>
> CASE type of
> 'UND': print, 'Could not evaluate datatype!'
> 'BYT': newarr=make_array(newdim,1, /BYTE)
> 'INT': newarr=make_array(newdim,1, /INT)
> 'LON': newarr=make_array(newdim,1, /LON)
> 'FLT': newarr=make_array(newdim,1, /FLOAT)
> 'DBL': newarr=make_array(newdim,1, /DOUBLE)
> 'COMPLEX': newarr=make_array(newdim,1, /COMPLEX)
> 'STR': newarr=make_array(newdim,1, /STRING)
> 'STC': newarr=replicate(oldarr(0),newdim)
> ENDCASE
>
> newarr(0:old_last)=oldarr(*)
>
> newarr(old_last+1:new_last) = newvals(*)
>
> return, newarr

More stuff deleted

>

Why bother with all of this when a single statement such as

newarr = [oldarr,newvals]

works just fine. It even works for structures (to my surprise). IDL already knows how to do all of these calculations so let it do it. If there are

problems with mixing variable types then the insertion of the proper function (fix, long, float, double, ...) should cure things. Also the occasional generation of a multidimensional array could be cured with the use of the (*) array subscripting. Seems like the previous post is doing alot of work for no good reason, or am I missing something obvious (afterall it is Friday morning and I have not had any coffee yet)?

--

Joseph M. Zawodny (KO4LW) NASA Langley Research Center
Internet: zawodny@arbd0.larc.nasa.gov MS-475, Hampton VA, 23681-0001
Packet: ko4lw@n4hog.va.usa