Subject: Re: resizing an array of structures (uugh)
Posted by Pavel A. Romashkin on Wed, 06 Jun 2001 17:01:48 GMT
View Forum Message <> Reply to Message

Hi Randall,

The experts answered already. I thought I might provide another way that I found much faster than anything else for files up to 100,000 rows. Read the file in a loop but use a large buffer, I use 10,000 rows. If the size of the buffer exceeds the length of the file, use Transfer_Count toobtain the right size for the buffer. Then, read the leftover data. Below is an excerpt from my own routine. It is not structure oriented but is easy to alter to fit your case:

```
; Read data from the file. Try to read by 10000 rows.
Buffer = fltarr(N_PrimVars+1) ; Because column 0 is always time.
readf, SourceFile, Buffer
; Create ResultArray with only 1 record, to initialize.
ResultArray = Transpose(Buffer)
; Create BIG buffer, in case file is long.
Buffer = fltarr(N_PrimVars+1, 10000)

; If Buffer was longer than file, reading will fail. In such case,
reduce buffer.
; Just to skip buffer reduction the first time,
FileTooShort = 0b
on_ioError, reduce_buffer
reduce_buffer : if FileTooShort eq 1b then begin
 unread_rows = (fstat(SourceFile)).Transfer_Count / (n_primVars+1)
 Buffer=fltarr(N_PrimVars+1, unread_rows)
 point_lun, SourceFile, LastPos
endif
; In case of ioError, this will be set to 1b
FileTooShort = 1b

while not EOF(SourceFile) do begin
; Memorize start in case reading fails.
 point_lun, (-SourceFile), LastPos
 readf, SourceFile, Buffer
 ResultArray = [ResultArray, Transpose(Buffer)]
endwhile
```