Subject: Re: Unsolved indexing problem 2 weeks ago.
Posted by Brian Larsen on Mon, 01 Oct 2007 19:37:36 GMT
View Forum Message <> Reply to Message

> While annoying, you only have to do it once.

All, I was writing a bunch of read_ascii wrappers today and got so annoyed at having to hard code in the ascii_template() output each time I wrote a code to do it for me...
It might be useful for you, and might not. You just run the code and then paste in the stuff between the SNIPS and that is your template for read_ascii().

[[Example]]

```
IDL> ascii_template_code
% Compiled module: ASCII_TEMPLATE_CODE.
% Compiled module: ASCII_TEMPLATE.
% Compiled module: QUERY_ASCII.
% Compiled module: XMANAGER.
;----------SNIP---------
template = create_struct('version',      1.00000, $
'datastart',          21l, $
'delimiter',      32b, $
'missingvalue', !values.f_nan, $
'commentsymbol', '', $
'fieldcount',          4l, $
'fieldtypes', lonarr(          4), $
'fieldnames', strarr(          4), $
'fieldlocations', lonarr(          4), $
'fieldgroups', lonarr(          4) )
template.fieldnames = [ $
'L', $
'PHI', $
'INTENSITY', $
'ERROR']
template.fieldtypes = [ $
          4, $
          4, $
          4, $
          4]
template.fieldlocations = [ $
          0, $
         15, $
         29, $
         43]
template.fieldgroups = [ $
          0, $
```

```
        1, $
        2, $
        3]
;----------SNIP---------
```

Cheers,

Brian

---------- ----------
Brian Larsen
Boston University
Center for Space Physics

--- code ----

```
;+
; NAME:
; ascii_template_code
;
;
; PURPOSE:
; put the code needed to hard code an ascii_template() in a read
routine
;
;
; CATEGORY:
; coding help
;
; INPUTS:
; none
;
;
; OPTIONAL INPUTS:
; file - the file to make a template of (opt since it will ask)
;
;
; KEYWORD PARAMETERS:
; none
;
;
; OUTPUTS:
; code printed to the screen
```

```
;
;
; OPTIONAL OUTPUTS:
; none
;
;
; COMMON BLOCKS:
; none
;
;
; SIDE EFFECTS:
; none
;
;
; RESTRICTIONS:
; limited testing, but has worked in each case I have tested on IDL6.4
;
;
;
; EXAMPLE:
; I cant come up with a simple one...
; IDL>  ascii_template_code
; template = create_struct('version',      1.00000, $
; 'datastart',           1l, $
; 'delimiter',      32b, $
; 'missingvalue', !values.f_nan, $
; 'commentsymbol', '', $
; 'fieldcount',          2l, $
; 'fieldtypes', lonarr(          2), $
; 'fieldnames', strarr(          2), $
; 'fieldlocations', lonarr(          2), $
; 'fieldgroups', lonarr(          2) )
; template.fieldnames = [ $
; 'PHI', $
; 'FIELD2']
; template.fieldtypes = [ $
;           4, $
;           4]
; template.fieldlocations = [ $
;           6, $
;          18]
; template.fieldgroups = [ $
;           0, $
;           1]
;
;
;
; MODIFICATION HISTORY:
```

```
;
;       Mon Oct 1 15:26:51 2007, Brian Larsen
;       <balarsen@bu.edu>
;
;  written and tested
;
;-




pro ascii_template_code

temp = ascii_template()
print, ';----------SNIP---------'
print, "template = create_struct('version'," + string(temp.version) +
', $'
print, "'datastart'," + string(temp.datastart)+'l, $'
print, "'delimiter'," + string(fix(temp.delimiter))+'b, $'
if not finite(temp.missingvalue) then $
  print, "'missingvalue', !values.f_nan, $" $
else $
  print, "'missingvalue'," + string(temp.missingvalue) + ', $'
print, "'commentsymbol', '"+ temp.commentsymbol + "', $"
print, "'fieldcount'," + string(temp.fieldcount) + 'l, $'
print, "'fieldtypes', lonarr("+string(temp.fieldcount)+'), $'
print, "'fieldnames', strarr("+string(temp.fieldcount)+'), $'
print, "'fieldlocations', lonarr("+string(temp.fieldcount)+'), $'
print, "'fieldgroups', lonarr("+string(temp.fieldcount)+') )'
print, "template.fieldnames = [ $"
for i=0l, temp.fieldcount-2  do $
  print, "'" + temp.fieldnames[i] + "', $"
print,  "'" + temp.fieldnames[i] + "']"
print, "template.fieldtypes = [ $"
for i=0l, temp.fieldcount-2  do $
  print, string(temp.fieldtypes[i]) + ", $"
print,  string(temp.fieldtypes[i]) + "]"

print, "template.fieldlocations = [ $"
for i=0l, temp.fieldcount-2  do $
  print, string(temp.fieldlocations[i]) + ", $"
print,  string(temp.fieldlocations[i]) + "]"

print, "template.fieldgroups = [ $"
for i=0l, temp.fieldcount-2  do $
  print, string(temp.fieldgroups[i]) + ", $"
print,  string(temp.fieldgroups[i]) + "]"
```

```
print, ';----------SNIP---------'



return
end
```