Subject: system routines
Posted by Martin Schultz on Fri, 14 Jan 2000 08:00:00 GMT
View Forum Message <> Reply to Message

Hi all,

while waiting for the fortran compiler to finish this model (ahgrr - the

next error ...) I played a little with IDL. On my backburner I have plans to revise my IDL library and create a website with my routines and others which shall be much more comfortable than what I had before (Reimar will know what I have in mind ;-). Of course, the building process has to be automated...

One neat feature that I would like to implement is a list of procedure or function dependencies. This is where I started playing today. Please find attached routines.pro which contains create_routines: will produce IDL_ROUTINES-<release> listing all system procedures and functions routines: call as routines, filename - will search the file filename and print all

occurences of system procedures and functions
While I am pretty sure to catch all procedures (trick is they must be
at the beginning of a line), there are certainly some more involved
issues with functions which I want to deal with later. Right now, what
interests me is if someone knows a good way to get the reserved words in
IDL other than copying and pasting from the online help. If not, then
that's what I will have to do hoping that there weren't too many changes
over the last century;-)

Enjoy! Martin

PS: if the shareware experiment will achieve some success, I might distribute the complete webpage generator in this form. Sounds appealing if you sit on the author's side ;-)

; program to create list of system routines and

```
; small test program to detect system routines in user code
; Martin Schultz, 14 Jan 2000
pro create_routines
 ; will search for all system procedures and functions and store them
  ; as IDL ROUTINES-<release>
 listname = 'IDL ROUTINES-'+!version.release
 open file, listname, llun, /WRITE
 if Ilun le 0 then begin
   message, 'Cannot open output file '+listname, /Continue
 endif
 printf,llun,'# list of internal procedures and functions in IDL ' + $
     !version.release
 printf, llun, '# obtained with routine_info(/SYSTEM [,/FUNCTIONS])'
 printf, llun, '[PROCEDURES]'
 n = routine info(/SYSTEM)
 for i=0L,n elements(n)-1 do printf, llun, n[i]
 printf, Ilun, '[FUNCTIONS]'
 n = routine_info(/SYSTEM,/FUNCTIONS)
 for i=0L,n_elements(n)-1 do printf,llun,n[i]
 free_lun,llun
 print, 'Done.'
return
end
pro routines, filename
 MAXNAMES = 2000
 pnames = strarr(MAXNAMES)
 fnames = strarr(MAXNAMES)
 ; get list of IDL routines and functions first
 listname = 'IDL ROUTINES-'+!version.release
 open file, listname, llun
 if Ilun le 0 then begin
   message, 'Cannot open routine list '+listname, /Continue
   return
 endif
 np = 0L
 nf = 0L
 s = "
```

```
state = 0; 1 = proc, 2 = func
 while (not eof(Ilun) AND np lt MAXNAMES $
      AND of It MAXNAMES) do begin
   readf,llun,s
   s = strtrim(s,2)
   if s eq '[PROCEDURES]' then state=1
   if s eq '[FUNCTIONS]' then state=2
   if strpos('#[!',strmid(s,0,1) ) It 0 then begin
     case state of
       1: begin
         pnames[np] = s
         np = np + 1
         end
       2: begin
         fnames[nf] = s
         nf = nf + 1
         end
      else:
     endcase
   endif
 endwhile
 free lun, llun
 pnames = pnames[0:np-1]
 fnames = fnames[0:nf-1]
help,pnames,fnames
 open file, filename, ilun
 if ilun le 0 then return
 s = "
 while not eof(ilun) do begin
   readf,ilun,s
   ; trim blanks for procedure names always start on new line
   s = strcompress(s,/remove all)
   ; look for name of procedure [perhaps followed by ',']
   test = strupcase( (str_sep(s,','))[0] )
   found = where(pnames eq test)
   if found[0] ge 0 then print, '** found ',pnames[found[0]] $
   else begin
     ; look for function after '=' (assuming there is no line break)
     ; make sure it's not commented out
     p = strpos(s,'=')
     pc = strpos(s,';')
     if (p gt 0 AND p gt pc) then begin
       p1 = strpos(s, '(', p+1))
       if p1 ge p then begin
```

```
test = strupcase(strmid(s,p+1,p1-p-1))
; print,test
          found = where(fnames eq test)
          if found[0] ge 0 then print, "## found ', fnames[found[0]]
       endif; else could be a '$' or ';' following ...
      endif
    endelse
  endwhile
 free_lun,ilun
return
end
```

File Attachments

1) routines.pro, downloaded 75 times