

---

Subject: Two new tools

Posted by [Martin Schultz](#) on Fri, 27 Mar 1998 08:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Howdy,

today I finally got around improving my concept of displaying help information on self-written (and library!) routines. Please find attached

function routine\_name.pro - which returns the name of the caller routine  
and

procedure usage.pro - which displays the "user-relevant" portion of the IDL program header (with minor modifications).

The idea of these routines is to prevent wrist injury from too much double typing in constructs like:

```
if (keyword_set(help)) then begin
    print ; all the stuff that's in the header anyway
endif
```

Now, what you can do is:

```
if (keyword_set(help)) then usage,routine_name()
and you will get a significant portion from the program header
[at least if your routine is stored in a file with the same name].
You can also use usage to get help from the command line:
usage,'my_routine_with_too_many_parameters'
```

Hope you'll find this useful,  
Martin.

--

-----  
Dr. Martin Schultz  
Department for Earth&Planetary Sciences, Harvard University  
186 Pierce Hall, 29 Oxford St., Cambridge, MA-02138, USA

phone: (617)-496-8318  
fax : (617)-495-4551

e-mail: [mgs@io.harvard.edu](mailto:mgs@io.harvard.edu)  
IDL-homepage: <http://www-as.harvard.edu/people/staff/mgs/idl/>

-----  
;-----  
;+  
; NAME:  
; ROUTINE\_NAME (function)

```

;
; PURPOSE:
;   return the name of the routine which calls this function.
;
; CATEGORY
;   Tools
;
; CALLING SEQUENCE:
;   rname = ROUTINE_NAME()
;
; INPUTS:
;   none
;
; KEYWORD PARAMETERS:
;   none
;
; OUTPUTS:
;   The name of the caller routine is returned in lowercase
;   characters (can be used to construct a filename by adding
;   ".pro")
;
; SUBROUTINES:
;
; REQUIREMENTS:
;
; NOTES:
;
; EXAMPLE:
;   From the command line:
;     print,routine_name()
;   results in $main$
;
;   Very useful in conjunction with USAGE.PRO:
;     usage,routine_name()
;   displays help information on the current routine.
;
; MODIFICATION HISTORY:
;   mgs, 27 Mar 1998: VERSION 1.00
;
;-
; Copyright (C) 1998, Martin Schultz, Harvard University
; This software is provided as is without any warranty
; whatsoever. It may be freely used, copied or distributed
; for non-commercial purposes. This copyright notice must be
; kept with any copy of this software. If this software shall
; be used commercially or sold as part of a larger package,
; please contact the author to arrange payment.
; Bugs and comments should be directed to mgs@io.harvard.edu

```

```
; with subject "IDL routine routine_name"  
;----- --
```

```
function routine_name
```

```
    ; extract the name of the current routine from the caller stack  
    ; the first element will always be ROUTINE_NAME ;-)  
    help,call=c
```

```
    thisroutine = str_sep(c(1)," ")
```

```
return,strlowercase(thisroutine(0))  
end
```

```
;----- --
```

```
;+
```

```
; NAME:
```

```
;   USAGE
```

```
;
```

```
; PURPOSE:
```

```
;   Display help information on any routine in the IDL path  
;   that has a (more or less) standard header.
```

```
;
```

```
; CATEGORY:
```

```
;   Tools
```

```
;
```

```
; CALLING SEQUENCE:
```

```
;   USAGE,routinename
```

```
;
```

```
; INPUTS:
```

```
;   ROUTINENAME -> (string) name of the routine for which help information  
;   shall be provided. Tip: to get help for the current routine use  
;   function routine_name().
```

```
;
```

```
; KEYWORD PARAMETERS:
```

```
;   /PRINTALL -> prints complete header information. Normally, only  
;   "user relevant" information is displayed.
```

```
;
```

```
; OUTPUTS:
```

```
;   prints usage information on the screen.
```

```
;
```

```
; SUBROUTINES:
```

```
;
```

```
; REQUIREMENTS:
```

```
;
```

```

; NOTES:
;   This routine is meant to replace /HELP constructs etc.
;
; EXAMPLE:
;   Error checking :
;
;       if (n_params() ne 2) then begin
;         print,'Invalid number of arguments!'
;         usage,routine_name()
;       endif
;
;   Get the idea what to type from the command line:
;       usage,'my_routine_with_too_many_arguments'
;
; MODIFICATION HISTORY:
;   mgs, 27 Mar 1998: VERSION 1.00
;
;-----
; Copyright (C) 1998, Martin Schultz, Harvard University
; This software is provided as is without any warranty
; whatsoever. It may be freely used, copied or distributed
; for non-commercial purposes. This copyright notice must be
; kept with any copy of this software. If this software shall
; be used commercially or sold as part of a larger package,
; please contact the author to arrange payment.
; Bugs and comments should be directed to mgs@io.harvard.edu
; with subject "IDL routine usage"
;-----

```

```

pro usage,rname,printall=printall

```

```

; extract header information from current routine and print it
; restrict to CALLING_SEQUENCE, INPUTS, KEYWORDS, OUTPUTS and
; EXAMPLE(s)
; This requires that the routine name is identical with the
; filename (lowercase) and that the file contains a standard
; header (i.e. mgs standard).

```

```

ON_ERROR,2 ; return to caller

```

```

; routine name must be provided!
if (n_params() eq 0) then $
  message,'pro USAGE: must provide routinename !'

```

```

; check if pro file exists
rfile = rname + '.pro'

```

```

if (not(file_exist(rfile,path=!PATH,full=full))) then $
  message,'pro USAGE: Cannot find program file for '+rname+' !'

; open file and read in header (read until line is ";" or EOF)
ilun = -1
ON_IOERROR,badfile

openr,ilun,full,/get_lun

line = "
if (keyword_set(PRINTALL)) then printit = 1 else printit = 0

print,strupcase(rname),':'
print

while (not (eof(ilun) OR line eq ';-' ) ) do begin
  readf,ilun,line

  ; determine whether to switch printing on or off
  teststr = strupcase(line)
  if (strpos(teststr,'CALLING SEQUENCE') ge 0 OR $
    strpos(teststr,'EXAMPLE') ge 0) then printit = 1
  if (strpos(teststr,'SUBROUTINE') ge 0 OR $
    strpos(teststr,'MODIFICATION') ge 0) then $
    if (not keyword_set(PRINTALL)) then printit = 0

  ; output header line if requested
  if (printit) then print,line
endwhile

close,ilun

return

badfile:
  if (ilun ge 0) then close,ilun
  print,!error,' ',!err_string
  message,'pro USAGE: File error in '+rfile+' !'

end

```

## File Attachments

---

- 1) [routine\\_name.pro](#), downloaded 149 times
  - 2) [usage.pro](#), downloaded 128 times
-