Subject: Which like command for IDL?
Posted by Jason P. Meyers on Fri, 05 Jan 2001 00:05:31 GMT
View Forum Message <> Reply to Message

Happy New Year All,

I was recently working with some examples from Dave Fanning's book (nice book Dave) when I realized it would be nice to have an IDL program that works like the unix "which" command. I had typed in the Display.pro program on page 66 only to find out that there is some other program (I suspect an RSI demo) also called Display.pro elsewhere on my IDL path. It would sure be nice to be able to quickly identify which program IDL is using from the path.

Does anyone out there know of such a beast for IDL?

Thanks in advance,
Jason Meyers
PhD Student, Center for Imaging Science
Rochester Institute of Technology
jpm7934@rit.edu

Subject: Re: Which like command for IDL?
Posted by William Thompson on Mon, 08 Jan 2001 15:47:38 GMT
View Forum Message <> Reply to Message

Here's a which program which a colleague of mine (Stein Vidar Haugan) wrote. It's very good. A particularly nice feature is that one can do "which,/all" to show all the matches in the path. Any routines which it may call, such as concat_dir.pro, can be found in one of the subdirectories of

ftp://sohoftp.nascom.nasa.gov/solarsoft/gen/idl/

William Thompson

"Jason P. Meyers" wrote:

>
> Happy New Year All,
>
> I was recently working with some examples from Dave Fanning's book
> (nice book Dave) when I realized it would be nice to have an IDL program
> that works like the unix "which" command. I had typed in the
> Display.pro program on page 66 only to find out that there is some other
> program (I suspect an RSI demo) also called Display.pro elsewhere on my
> IDL path. It would sure be nice to be able to quickly identify which

> program IDL is using from the path.
>
>   Does anyone out there know of such a beast for IDL?
>
> Thanks in advance,
> Jason Meyers
> PhD Student, Center for Imaging Science
> Rochester Institute of Technology
> jpm7934@rit.edu

```
PRO WHICH, name, all=all, search=search, outfile=outfile,quiet=quiet
;+
; PROJECT:
;      SOHO - CDS
;
; NAME:
;      WHICH
;
; PURPOSE:
;      Search for and print file or routine in IDL !path
;
; EXPLANATION:
;      Use to find where IDL finds a program file, and, if
; multiple definitions exist, which is loaded (the first
; one). Splits path into all different directories,
; searches for a file with the given NAME + '.PRO'.
;
; CALLING SEQUENCE:
;      WHICH, NAME
;
; INPUTS:
;      NAME - Name of the routine to search for (string scalar).
;
; OPTIONAL INPUTS:
;      None.
;
; OUTPUTS:
;      None.
;
; OPTIONAL OUTPUTS:
;      outfile - return file name found
;
; KEYWORD PARAMETERS:
;      ALL    - Report all occurrences if set. Usually WHICH reports
the
;              first occurrence of a found routine (which is to be
;              executed by IDL). WHICH can take a while, especially on
;               VMS system, to search through all directories (and text
;               libraries on VMS system) if ALL is set. ALL is
;               automatically set if SEARCH is set.
;       SEARCH - Turn on the search mode, if set, that would match any
;               given string pattern found in the path.
;
; CALLS:
;       CONCAT_DIR, STR_CHOP, GREP
;
; COMMON BLOCKS:
;       WHICH -- Mainly for speeding things up
;
; RESTRICTIONS:
;       None.
;
; SIDE EFFECTS:
;       None.
;
; CATEGORY:
;       General utility
;
; PREVIOUS HISTORY:
;       Written Stein Vidar Haugan, 1993
;
; MODIFICATION HISTORY:
;       19 May, 1994, SVHH, Doc. added
; 21 May, 1994, SVHH, Version 2, with on_error,2 and 'Use:'
;       Liyun Wang, GSFC/ARC, September 20, 1994
;          Added IDL internal routine checkings.
;       Liyun Wang, GSFC/ARC, October 5, 1994
;          Current directory also gets searched now
;       Version 3, Liyun Wang, GSFC/ARC, December 16, 1994
;          Made it capable of finding files in text libraries on VMS
system
;          Added the ALL keyword
;       Version 4, Liyun Wang, GSFC/ARC, January 23, 1995
;          Added the SEARCH keyword
;       Version 5, Liyun Wang, GSFC/ARC, January 24, 1995
;          Used an undocumented IDL function routine ROUTINE_NAMES to
get IDL
;             intrinsic routine names.
;       Version 6, Liyun Wang, NASA/GSFC, October 1, 1996
;          Added Windows support
;       Version 7 Add output variable.  CDP, RAL 14-Mar-97
; Version 8, 23-Oct-1997, William Thompson, GSFC
;  Use OS_FAMILY() instead of !VERSION.OS
; Version 9, 4-July-1998, Zarro (SAC/GSFC)
;              Returned scalar out variable for single element value
;       Version 10, 28-Jun-1999, Zarro (SM&A/GSFC)
;               Added /quiet
;
; VERSION:
;       Version 9
;-

  COMMON which, internal
  ON_ERROR,2
  loud=1-keyword_set(quiet)

 ;---------------------------------------------------------- -----------
;  A list of IDL internal routine can be obtained by entering the "?"
;  command at IDL prompt, provided that the current device is set to
;  'tek'.
 ;---------------------------------------------------------- -----------

  IF N_ELEMENTS(name) EQ 0 THEN BEGIN
    PRINT, 'WHICH -- Syntax error.'
    PRINT, '   Usage: WHICH, "filename" [,/ALL, file=file]'
    PRINT, ' '
    RETURN
  ENDIF

;
;  keep an array of the answers
;
outfile = ''

 ;---------------------------------------------------------- -----------
;  In case a file name with an extension is entered, strip off the
;  extension:
 ;---------------------------------------------------------- -----------
  ipos =  STRPOS(name,'.')
  IF (ipos NE -1) THEN name = STRMID(name,0,ipos)
  cd,current = cur_dir

  CASE os_family() OF
    'vms': path_sep = ','
    'Windows': path_sep = ';'
    ELSE: path_sep = ':'
  ENDCASE

 ;---------------------------------------------------------- -----------
;  First check to see if it is an IDL internal routine. We will call an
;  undocumented IDL function called ROUTINE_NAMES to get all names of
;  the IDL built-in function and procedure names.
 ;---------------------------------------------------------- -----------
```

```
  IF N_ELEMENTS(internal) EQ 0 THEN BEGIN
    f_name = ROUTINE_NAMES(/s_functions)
    p_name = ROUTINE_NAMES(/s_procedures)
    internal = [f_name, p_name]
  ENDIF
  IF KEYWORD_SET(search) THEN BEGIN
    exact = 0
    all = 1
    ff = grep(name, internal)
    IF ff(0) NE '' THEN BEGIN
      temp = arr2str(ff,', ')
      if loud then PRINT, 'IDL Built-in: ',temp
      outfile = [outfile,'IDL Built-in: '+temp]
    ENDIF
  ENDIF ELSE BEGIN
    exact = 1
    aa = WHERE(STRUPCASE(name) EQ internal, count)
    IF count NE 0 THEN BEGIN
      if loud then PRINT, STRUPCASE(name)+' is an IDL built-in
routine.'
      outfile = [outfile,STRUPCASE(name)+' is an IDL built-in
routine.']
      IF N_ELEMENTS(all) EQ 0 THEN begin
        if n_elements(outfile) gt 1 and outfile(0) eq '' then $
          outfile = outfile(1:*)
        RETURN
      ENDIF
    ENDIF
  ENDELSE

  p = cur_dir+path_sep+!path
  dirs = str_sep(p,path_sep)

  CASE (!version.os) OF
    'vms': BEGIN
      FOR i = 0,N_ELEMENTS(dirs)-1 DO BEGIN
        allfiles = str_chop(STRUPCASE(get_mod(dirs(i))),'.PRO')
        ff = grep(name, allfiles, exact=exact)
        IF (ff(0) NE '') THEN BEGIN
          FOR j = 0, N_ELEMENTS(ff)-1 DO BEGIN
            filename = concat_dir(dirs(i),ff(j))
            if loud then PRINT, filename+'.PRO'
            outfile = [outfile,filename+'.PRO']
          ENDFOR
          IF N_ELEMENTS(all) EQ 0 THEN GOTO, cleanup
        ENDIF
      ENDFOR
cleanup:
 ;-------------------------------------------------------- -----------
;     Clean up scratch files when library is listed
 ;-------------------------------------------------------- -----------
      look = findfile('sys$login:*._sdac_*;*',count=nf)
      IF nf GT 0 THEN BEGIN
        FOR i=0,nf-1 DO BEGIN
          spawn,'DELETE/NOLOG/NOCONFIRM '+look(i)
        ENDFOR
      ENDIF
    END
    ELSE: BEGIN
      IF KEYWORD_SET(search) THEN BEGIN
        FOR i = 0, N_ELEMENTS(dirs)-1 DO BEGIN
          filename =
findfile(concat_dir(dirs(i),'*'+name+'*.pro'),$
                  count = cnt)
          IF cnt GT 0 THEN BEGIN
            FOR j = 0, cnt-1 DO begin
              if loud then PRINT, filename(j)
              outfile = [outfile,filename(j)]
            endfor
          ENDIF
        ENDFOR
      ENDIF ELSE BEGIN
        FOR i = 0,N_ELEMENTS(dirs)-1 DO BEGIN
          filename = concat_dir(dirs(i),name+'.pro')
          OPENR, unit, filename, /GET_LUN, error=rr
          IF rr EQ 0 THEN BEGIN
            if loud then PRINT, filename
            outfile = [outfile,filename]
            CLOSE, unit
            FREE_LUN, unit
            IF NOT KEYWORD_SET(all) THEN BEGIN
              if n_elements(outfile) gt 1 and outfile(0) eq ''
then $
                 outfile = outfile(1:*)
              RETURN
            ENDIF
          ENDIF
        ENDFOR
      ENDELSE
    END
  ENDCASE
;
;  tidy up output variable
;
if n_elements(outfile) gt 1 and outfile(0) eq '' then outfile =
outfile(1:*)
```

```
if n_elements(outfile) eq 1 then outfile=outfile(0)

END
```

Subject: Re: Which like command for IDL?
Posted by R.Bauer on Fri, 12 Jan 2001 14:09:00 GMT
View Forum Message <> Reply to Message

"Jason P. Meyers" wrote:
>
> Happy New Year All,
>
> I was recently working with some examples from Dave Fanning's book
> (nice book Dave) when I realized it would be nice to have an IDL program
> that works like the unix "which" command. I had typed in the
> Display.pro program on page 66 only to find out that there is some other
> program (I suspect an RSI demo) also called Display.pro elsewhere on my
> IDL path. It would sure be nice to be able to quickly identify which
> program IDL is using from the path.
>
> Does anyone out there know of such a beast for IDL?
>

Dear Jason,

We have a general file search routine in our library.

```
; NAME:
;   file_search
;
; PURPOSE:
;   This function searches a file in the current directory or in a list
of directories
;   This function does return the first file which was found to match
the search
;   criterion. Use the wild cards '*' or /LIST keyword to return a full
list.
;
; CATEGORY:
;   PROG_TOOLS/FILES
;
; CALLING SEQUENCE:
;   Result=FILE_SEARCH(Name)
;
; INPUTS:
;   Name:   name of a file or wildcards
```

```
;
; KEYWORD PARAMETERS:
;   DIRECTORIES:    string (array) containing a list of directories to
be searched
;   NO_CURRENT:     if set the current directory is not searched !
;   ENV_VAR:        name of a environment variable containing a path
;   USE_PATH:       if this keyword is set the path of IDL is used for
search
;   SEARCH_TREE:    if this keyword is set the search is expanded to all
subdirs of the
;                   given directories, equivalent to a preceeding '+' in
the directory name
;   FILE_COUNT:     if set to named variable the number of file hits is
returned
;   DIR_COUNT:      if set to a named variable the number searched
directories is returned
;   DIR_FOUND:      if set to a named variable a string array of
searched dirs is returned
;   RETURN_MESSAGE: if set to a named variable the message is returned
```

http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_source/idl_ html/dbase/download/file_search.tar.gz

For further routines and licensing please look at

http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_lib_intro.h tml

regards
Reimar

--
Reimar Bauer

Institut fuer Stratosphaerische Chemie (ICG-1)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de