

---

Subject: FINDFILE

Posted by [J.D. Smith](#) on Mon, 07 Jul 1997 07:00:00 GMT

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

---

I'd like to register a complaint with the Disgruntled-IDL-Programmers-of-the-World. It may seem silly, but it's caused me great headaches. Findfile() does not perform the same way cross-platform. On unix machines, the command findfile() finds all files \*and\* directories at the current level, but does not distinguish between the two. On windows/mac, you get a separator for directories, allowing easy parsing. Also, findfile('\*') on unix descends one level to list all files and directories through that level, but still with inconsistent directory demarcation. It would be so lovely if findfile() produced the list of current files/directories at the current level, with a platform specific separator following the directories. Please, RSI, make my life a little easier...

---

---

Subject: Re: findfile

Posted by [David Foster](#) on Fri, 28 May 1999 07:00:00 GMT

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

---

R.Bauer wrote:

>  
> Hi,  
>  
> I got in trouble by findfile on a unix sytem  
>  
> while Windows IDL returns by findfile('C:\\*.\*') all files  
> unix (aix) IDL did not give a result if more than 3500 files in a  
> directory.  
> findfile('/tmp/\*.\*') is "  
>  
> If I use findfile('/tmp') I got all files.  
>  
> Unfortunately I have momentanly on the unix only idl5.1  
>  
> What is idl5.2 doing ?  
>  
> R.Bauer

I thought I'd post yet another version of a workaround for this. The FILE\_FIND.PRO below uses 'ls' on UNIX platforms, and you can use the /DIR keyword to find directories, or the /RECURSIVE keyword to make the search recursive, on UNIX systems.

I've also attached the doc file FILE\_FIND.DOC.

Dave Foster

--

```
~~~~~  
David S. Foster      Univ. of California, San Diego  
Programmer/Analyst  Brain Image Analysis Laboratory  
foster@bial1.ucsd.edu  Department of Psychiatry  
(619) 622-5892      8950 Via La Jolla Drive, Suite 2240  
                    La Jolla, CA 92037  
~~~~~
```

```
; FILE_FIND.PRO 10-29-98 DSFoster  
;  
;-----  
; function File_Find  
;  
; Takes file specification, expands it and returns a list of matching  
; filenames. Returns -1 if no files are found (and argument FNAMES will  
; be undefined). Returns -2 if arg FILESPEC is undefined or null.  
; Returns the number of files found if no errors. Mostly borrowed from  
; the PICKFILE widget.  
;  
; Set the DIR keyword if you want only the names of directories returned.  
; In this case, FILESPEC has to specify a directory itself.  
;  
; Set the RECURSIVE keyword to have all files or directories searched  
; recursively down the directory tree (Unix only).  
;  
; Modifications:  
;  
; 1-12-95 DSF Creation.  
; 2-04-97 DSF Sort the filenames before returning. Set the keyword  
; NO_SORT to disable this.  
; 6-13-97 DSF Make /RECURSIVE and /DIR independent, so you can search  
; recursively for files as well as directories. For non-UNIX  
; platforms just call FINDFILE() (/DIR and /RECURSIVE not  
; allowed).  
; 10-29-98 DSF Use "find" command for /RECURSIVE to get correct results!  
; Add /PATH for use with /RECURSIVE to specify directory to  
; begin search. If /DIR specified use "ls" and return only  
; directory names. Add "-d" argument to "ls" command for  
; nonrecursive searches to avoid listing directory contents.  
;-----
```

```
FUNCTION file_find, filespec, fnames, DIR=dir, RECURSIVE=recursive, $  
NO_SORT=no_sort, PATH=path
```

```

status = 0
fnames = "

if (n_elements(filespec) eq 0) then begin
  status = -2
endif else if (strlen(filespec) eq 0) then begin
  status = -2
endif else begin

if ( !version.os_family eq 'unix' ) then begin

  ON_IOERROR, io_error

; Use the FIND command for /RECURSIVE, and LS otherwise.

if (keyword_set(DIR)) then begin
  if (keyword_set(RECURSIVE)) then begin
    command = 'ls -lR ' + filespec + ' 2> /dev/null'
    cmd = 'ls'
  endif else begin
    command = 'ls -l ' + filespec + ' 2> /dev/null'
    cmd = 'ls'
  endelse
endif else if (keyword_set(RECURSIVE)) then begin
  command = 'find ' + filespec + ' -type d 2> /dev/null'
  cmd = 'find'
endif else begin
  command = "ls -ld " + filespec + " 2> /dev/null"
  cmd = 'ls'
endelse

if (keyword_set(PATH)) then begin
  cd, current=curdir
  cd, path
endif

SPAWN, ["/bin/sh", "-c", command], results, /NOSHELL

if (keyword_set(PATH)) then $
  cd, curdir

if (keyword_set(RESULTS)) then begin
  if (keyword_set(DIR)) then begin
    firsts = strupcase(strmid(results, 0, 1))
    fileinds = where(firsts eq "D", found)
  endif else if (keyword_set(RECURSIVE)) then begin
    pos = strpos(results, './')

```

```

ind = where(pos eq 0)
results(ind) = strmid(results(ind), 2, 1000)
fnames = results
found = (status = n_elements(fnames))
endif else begin
firsts = strupcase(strmid(results, 0, 1))
fileinds = where(firsts eq "F" or firsts eq "-" OR $
  firsts eq "I", found)
endelse
if (found GT 0) then begin
if (cmd eq 'ls') then begin
results = results(fileinds)
FOR i = 0, n_elements(results) - 1 DO begin
spaceinds = where(BYTE(results(i)) EQ 32)
spaceindex = spaceinds(n_elements(spaceinds) - 1)
results(i) = strmid(results(i), spaceindex + 1, 100)
endFOR
fnames = results
if (not keyword_set(NO_SORT)) then $ ; Sort?
fnames = fnames( sort(fnames) )
status = n_elements(fnames) ; Return-value
endif
endif else begin
status = -1
endelse
endif else begin
status = -1
endelse

endif else begin ; Non-UNIX platforms

if (keyword_set(DIR) or keyword_set(RECURSIVE)) then begin
message, 'Keywords DIR and RECURSIVE on UNIX platforms only'
endif else begin
fnames = findfile(filespec, count = status)
if (fnames(0) eq "") then $
status = -1
endelse

endelse

endelse

return, status

io_error: return, -1

end

```

## FILE\_FIND

Takes a file specification string, probably containing wildcards, and returns a sorted array of matching filenames as the second argument. Returns the number of files found if no error.

### Calling Sequence

```
Found = FILE_FIND(Filespec, Filenames)
```

### Arguments

#### Filespec

A string representing the file-specification to match. All files matching this Filespec will be found. Type: STRING.

#### Filenames

The list of filenames matching the Filespec. Type: STRARR (an array of strings). Note that this argument will be undefined if an error is encountered or no files are found. This list of filenames is NOT sorted. (See example below.)

### Keywords

#### DIR

Set this when you want directories returned. Note that Filespec must specify a directory itself. Only the directory names are returned, not the paths.

ONLY ON UNIX SYSTEMS.

#### NO\_SORT

The default behavior is to sort the array of filenames to return. Set this keyword to prevent this.

#### PATH

Use this with the RECURSIVE keyword to specify the directory location where the recursive search should begin. The default is the current directory.

## RECURSIVE

Search for files having names matching Filespec recursively in all subdirectories. Returns the complete pathname for each file found.

ONLY ON UNIX SYSTEMS.

## Outputs

Fills argument Filenames with the list of filenames found. If an error is encountered or no files are found, Filenames will be undefined.

Returns the number of files found. If no files are found returns -1, and -2 if Filespec argument is undefined or null.

## Example

```
found = FILE_FIND('/dir/im/*.doc', fnames)

; Now sort Fnames
fnames = fnames(SORT(fnames))
```

## File Attachments

---

- 1) [file\\_find.pro](#), downloaded 129 times
  - 2) [file\\_find.doc](#), downloaded 106 times
- 

---

Subject: Re: findfile  
Posted by [gurman](#) on Fri, 28 May 1999 07:00:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

In article <374E8944.A85478F3@fz-juelich.de>, "R.Bauer" <R.Bauer@fz-juelich.de> wrote:

```
> Hi,
>
> I got in trouble by findfile on a unix sytem
>
> while Windows IDL returns by findfile('C:\*.*) all files
> unix (aix) IDL did not give a result if more than 3500 files in a
```

```

> directory.
> findfile('/tmp/*.*) is "
>
> If I use findfile('/tmp') I got all files.
>
> Unfortunately I have momentanly on the unix only idl5.1
>
> What is idl5.2 doing ?
>
>
> R.Bauer

```

As several people have noted, Bill Thompson solved this for the solarsoft library.

Joe Gurman

```

;+
; Project   : SOHO - CDS
;
; Name      : FIND_FILE()
;
; Purpose   : Fixing builtin FINDFILE() problem
;
; Explanation : The builtin FINDFILE() function has problems on some unices
;               whenever *a lot* of files are matching the file
;               specification. This is due to the fact that filename expansion
;               is done by the shell *before* interpreting a command. Too many
;               files cause too long commands, which are not accepted. This
;               causes FINDFILE() to return an empty list of candidates.
;
;               FIND_FILE tries the builtin function first, and whenever the
;               returned list of files is empty, it tries to recheck through
;               spawning a "find" command.
;
;               Since FINDFILE doesn't discriminate between directories, links
;               and files, this function will not do this either.
;
;               Under unix, however, calls like FINDFILE("") returns the
;               unfiltered output of the shell commmand "ls **", including
;               colon-terminated lines for each subdirectory matching the
;               specification and empty lines separating each subdirectory
;               listing. Such silly effects are not implemented in the "find"
;               version. Be warned, however, that these effects are present
;               when the builtin function does not "fail" due to a too long
;               file list.
;
;               It is possible (under unix) to use the "find" method as

```

```

;      default by setting the keyword /USEFIND (no effect under other
;      operating systems).
;
;
; Use      : files = find_file(file_specification)
;
; Inputs   : file_specification : A scalar string used to find
;           files. See FINDFILE()
;
;
; Opt. Inputs : None.
;
; Outputs   : Returns a list of files or a blank string if none found.
;
; Opt. Outputs:
;
; Keywords  : COUNT : Returns the number of files
;
;           USEFIND : Always use a spawned "find" command under unix.
;           No effect under other operating systems.
;
;           NODOT : Apply a filter to the results from find to prevent
;           finding the directory itself in a large file expansion.
;           eg 'find_file,"foo/*"' returns ("foo/","foo/a",...)
;           but 'find_file,"foo/*",/nodot' returns
;           ("foo/a","foo/b",...) without the leading "foo/".
;           This behavior is closer to the behavior of findfile()
;           without the long-directory braindamage. It is
;           *not* the default so as not to break heritage
;           code that uses find_file().
;
; Calls     : FINDFILE, SPAWN
;
; Common    : None
;
; Restrictions: As for FINDFILE
;
; Side effects: None, hopefully
;
; Category  : Utilities, Operating_system
;
; Prev. Hist. : Lots of problems with FINDFILE is hopefully history.
;
; Written    : S.V.H. Haugan, UiO, 12 April 1996
;
; Modified   : Version 2, SVHH, 10 June 1996
;             Moved the CD,curr_path command to avoid
;             returns without resetting path.
;             Version 3, SVHH, 26 June 1996
;             Took away the -type f argument to find, added

```

```

;           /USEFIND keyword.
;       : Added /nodot keyword C. DeForest 9-August-1998
;
; Version   : 3, 26 June 1996
;-

```

```

FUNCTION find_file,file_specification,count=count,usefind=usefind,nod ot=nodot
count = 0
use_find = KEYWORD_SET(usefind) AND os_family() EQ 'unix'

```

```

IF NOT use_find AND N_PARAMS() EQ 0 THEN BEGIN
    result = findfile(count = count)
    RETURN,result           ; Unix doesn't have problems with this
END

```

```

IF N_PARAMS() EQ 0 THEN file_specification = '*'
IF file_specification EQ " THEN file_specification = '*'

```

```

IF NOT use_find THEN result = findfile(file_specification,count=count) $
ELSE count = 0

```

```

;; Check for problems

```

```

IF count EQ 0 AND os_family() EQ 'unix' THEN BEGIN
    file = file_specification
    break_file,file,disk,dir,filnam,ext

```

```

;; Check if directory exists
IF dir NE " THEN BEGIN
    IF (findfile(dir))(0) eq " THEN RETURN,"
END

```

```

;; Temporary switch to that directory
IF dir NE " THEN cd,dir,current=curr_path

```

```

IF filnam+ext EQ " THEN filnam = '*'

```

```

;; Find all matching
spn = ["find",".", "-name",filnam+ext, "-print"]
spawn,spn,result,/noshell

```

```

;; Switch back to original directory
IF dir NE " THEN cd,curr_path

```

```

IF result(0) EQ " THEN RETURN," ; None matching, return

```

```

;; Get rid of current-directory match, if necessary

```

```

if keyword_set(nodot) and result(0) eq '.' then $
result = result(1:n_elements(result)-1)

;; Chop off './'
result = STRMID(result,2,1000)

;; Chip out subdirectories (for some reason, the -prune option doesn't
;; work properly, so I have dropped using it).

ix = WHERE(STRPOS(result,'/') EQ -1,count)
IF count EQ 0 THEN RETURN,"

;; Put back the specified (not full) path
result = dir + result(ix)
END

RETURN,result

```

END

```

.....
; End of 'findfile.pro'.
.....

```

--

| Joseph B. Gurman, NASA Goddard Space Flight Center, Solar Physics  
| Branch, Greenbelt MD 20771 USA / Federal employees are still  
| prohibited from holding opinions while at work. Therefore, any  
| opinions expressed herein are somebody else's.

Subject: Re: findfile  
Posted by [davidf](#) on Fri, 28 May 1999 07:00:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

R.Bauer (R.Bauer@fz-juelich.de) writes:

- > I got in trouble by findfile on a unix sytem
- >
- > while Windows IDL returns by findfile('C:\*.\*') all files
- > unix (aix) IDL did not give a result if more than 3500 files in a
- > directory.
- > findfile('/tmp/\*.\*') is "
- >
- > If I use findfile('/tmp') I got all files.
- >
- > Unfortunately I have momentanly on the unix only idl5.1

I thought Bill Thompson had a fix for this problem. Can't seem to find the reference, however, in the places I usually save this kind of information (I.e, my web page). I'd check DejaNews for William Thompson articles. :-)

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting

Phone: 970-221-0438 E-Mail: davidf@dfanning.com

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Toll-Free IDL Book Orders: 1-888-461-0155

---

Subject: Re: findfile

Posted by [Thomas A. McGlynn](#) on Fri, 28 May 1999 07:00:00 GMT

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

---

This comes up periodically as one of IDL's quirks (to use a kind word). IDL seems to just use the Unix 'ls' command in findfile. You'll probably find that 'ls /tmp/\*.\*' would also fail. Unix expands the argument list before executing the command and the argument list overflows some limit.

The last time this came up a few months ago a couple of readers indicated that they had written replacements for findfile to get around this. You might check dejanews.

Regards,

Tom McGlynn

"R.Bauer" wrote:

>  
> Hi,  
>  
> I got in trouble by findfile on a unix sytem  
>  
> while Windows IDL returns by findfile('C:\*.\*') all files  
> unix (aix) IDL did not give a result if more than 3500 files in a  
> directory.  
> findfile('/tmp/\*.\*') is "  
>  
> If I use findfile('/tmp') I got all files.  
>  
> Unfortunately I have momentanly on the unix only idl5.1  
>



```

; MFINDFILE
;
;
; PURPOSE:
; find all the files that match a given specification.
; On our system, the IDL findfile function does not
; work correctly!!
;
; CATEGORY:
; System routines
;
; CALLING SEQUENCE:
; listing = MFINDFILE(filemask)
;
; INPUTS:
; FILEMASK -> a path and filename specification to look
; for.
;
; KEYWORD PARAMETERS:
; none
;
; OUTPUTS:
; A string list containing all the files that match the
; specification.
;
; SUBROUTINES:
;
; REQUIREMENTS:
;
; NOTES:
; Spawns a unix ls -l command !
;
; EXAMPLE:
; list = mfindfile('~mgs/terra/chem1d/code/*.f')
;
; ; returns all fortran files in Martin's chem1d directory.
;
; MODIFICATION HISTORY:
; mgs, 14 Sep 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 mfindfile"  
;-----
```

```
function mfindfile,mask  
  
if (!version.os_family eq 'unix') then begin  
; make my own findfile  
  
    path = extract_path(mask,filename=fname)  
    path = expand_path(path)  
    newpath = path+fname  
  
; print,'fname:',fname,' path:',path,' newpath:',newpath  
  
    command = 'ls -l'  
    cstr = command+ ' '+newpath  
    spawn,cstr,listing  
  
    return,listing  
  
endif else begin ; other OS - use IDL's original  
  
    return,findfile(mask)  
  
endelse  
  
end
```

## File Attachments

1) [mfindfile.pro](#), downloaded 119 times

---

---

Subject: Re: findfile

Posted by [R.Bauer](#) on Fri, 28 May 1999 07:00:00 GMT

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

---

"Martin LUETHI GL A8.1 2-4092" wrote:

> Hello

>

> I have a related question (concerning PV-Wave on a unix system): Is there a  
> way to expand the filename ~/my/path/to/data to the correct path in the call  
> to findfile()? This kind of filename works for all file commands but not for  
> findfile(). This a behaviour which is not very convenient since one has to use  
> absolute paths instead paths relative to the user home directory, causing

> severe problems with portability.  
>  
> Cheers  
>  
> Martin  
>

expand\_path() expands the path on idl

>  
> --  
> =====  
> Martin Luethi           Tel. +41 1 632 40 92  
> Glaciology Section      Fax. +41 1 632 11 92  
> VAW ETH Zuerich  
> CH-8092 Zuerich         mail luthi@vaw.baum.ethz.ch  
> Switzerland  
> =====

---

Subject: Re: findfile  
Posted by [luthi](#) on Fri, 28 May 1999 07:00:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

Hello

I have a related question (concerning PV-Wave on a unix system): Is there a way to expand the filename ~/my/path/to/data to the correct path in the call to findfile()? This kind of filename works for all file commands but not for findfile(). This a behaviour which is not very convenient since one has to use absolute paths instead paths relative to the user home directory, causing severe problems with portability.

Cheers

Martin

--  
=====

Martin Luethi   Tel. +41 1 632 40 92  
Glaciology Section   Fax. +41 1 632 11 92  
VAW ETH Zuerich  
CH-8092 Zuerich   mail luthi@vaw.baum.ethz.ch  
Switzerland

=====

Subject: Re: findfile

Posted by [thompson](#) on Sun, 30 May 1999 07:00:00 GMT

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

---

gurman@gsfc.nasa.gov (Joseph B. Gurman) writes:

> As several people have noted, Bill Thompson solved this for the  
> solarsoft library.

> Joe Gurman

Thanks for the plug Joe, but I only posted it. It was actually written by  
another familiar name on this newsgroup, Stein Vidar Haugan.

Bill

---