Subject: Re: existing directories
Posted by David Foster on Mon, 25 Jan 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Martin Schultz wrote:
>
> Hi,
>
> maybe something is blocking my mind right now, but I just can't think
> of a good (platform independent) way to test whether a directory exists
> or not. I am trying to write an 'install.pro' routine (not too fancy but
> at least with automatic backup of an old version and automatic creation
> of the target directory if not there). Actually, since I am spawning
> quite a few unix commands anyhow, a good Unix solution would be fine,
> too.

Martin -

Here are two routines that may help. First is a simple routine
that spawns a UNIX test command:

```
 ;------------------------------------------------------------ ------------------
;+
;   Function VALID_DIR
;
; Function to determine whether string value specifies valid directory.
; Returns 1 if valid, 0 if not.
;-
 ;------------------------------------------------------------ ------------------

FUNCTION valid_dir, dir

   ; Go to UNIX and check if directory exists using test command
   ; Open a new shell to avoid aliases

   spawn, ['/bin/sh -c "test -d ' + dir + '" && echo 1'], result

   return, keyword_set( result )
END
```

Here is DIREXIST.PRO which I got from Phil Williams some time ago:

```
---------- cut here -------------------------------------------
FUNCTION DirExist, list
;+
```

```
; NAME: DIREXIST
;
; PURPOSE: Determine which elements in list are directories.
;
; CATEGORY: File I/O
;
; CALLING SEQUENCE: result = direxist(list)
;
; INPUTS:
;    list : a list of files (i.e. the result from list = FINDFILE())
;
; OPTIONAL INPUTS: none
;
; OUTPUTS: An array of the appropriate size. 1 indicates list(i) is a
;          directory. 0 indicates that list(i) is a file.
;
; PROCEDURE:
;   Get a file list
;      IDL> list = findfile()
;      IDL> dirs = direxist(list)
;   Print which elements in list are directories
;      IDL> print,where(dirs eq 1)
;           1          6         14         22
;
; MODIFICATION HISTORY:
;   27 Dec 96 Initial coding. PMW
;     Use direxist.pro from David Fanning as a starting point.
;   28 Dec 96 Fixed bug. Now checking last entry in list. PMW
;   30 Dec 96 Fixed bug with last entry check. PMW
;-


; Save the current directory.

CD, Current=currentDirectory
foo = list
;print,'direxist dir = ', currentDirectory
; Use the Catch error handler to catch the case where we
; try to CD to a directory that doesn't exist.

results = intarr(n_elements(list))
i = -1
Catch, error
IF (error NE 0) THEN BEGIN

      ; Directory must not exist. Return 0.
    bar = 0
    goto, YIKES
```

```
ENDIF

   ; Try to CD to the directory. If it doesn't exist, an error occurs.

repeat begin
    i = i+1
    CD, foo(0), current = currentDirectory

; Well, the directory MUST exist if we are here! Change back to
; the current directory and return a 1.
    bar = 1

YIKES:
    results(i) = bar
    CD, currentDirectory
    if n_elements(foo) ne 1 then foo = foo(1:n_elements(foo) - 1)
endrep until n_elements(foo) eq 1
CATCH,/cancel

if n_elements(list) eq 1 then return, results
;--- Now do the last entry
i = i + 1

Catch, error
if (error ne 0) then begin

    results(i) = 0
    goto,done
endif

CD, foo(0), current = currentDirectory

;--- If were here its a directory
results(i) = 1

CD, currentDirectory

done:
RETURN, results
END



--

  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ ~~~~~~~
   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
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ ~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~