Subject: Logical links cause EXPAND_PATH() to hang
Posted by wlandsman on Tue, 26 Apr 2016 18:08:35 GMT
View Forum Message <> Reply to Message

I was trying to add a directory and all subdirectories to my !PATH under IDL 8.5 in Linux, by the usual method of adding a '+' in front of the directory name, e.g.
expand_path('+/users/wlandsma/pro')

But expand_path() would just hang, forcing me to kill the IDL session. I eventually tracked the problem to the presence of symbolic links. For example there was a 'pr3' link created with

ln -s /data2/wlandsma/pr3 pr3

Evidently this causes an infinite loop, as IDL keeps trying to expand the logical link. I think this is a bug. My current workaround is to manually add all 10 subdirectories.

Subject: Re: Logical links cause EXPAND_PATH() to hang
Posted by zacharyanorman on Thu, 28 Apr 2016 03:59:19 GMT
View Forum Message <> Reply to Message

I would contact technical support at support@harris.com to see if they can reproduce this problem. Otherwise, you can always recursively search each directory for files and only expand your path with folders.

Here is a function that will recursively find your subdirectories for you (I was curious how this would be done). It might take a few seconds depending on how many subfolders you have and I'm not sure if symbolic links will cause problems or not. Worked for me to recursively search my entire Dropbox folder which contains 998 folders total:

```
function expand_path_better, dir
  compile_opt idl2

  ;change directories to the first directory
  cd, dir, CURRENT = first_dir

  dirs = list()
  ;find the directories
  directories = file_search(/TEST_DIRECTORY, COUNT=ndirectories)

  ;check if we have no directories
  if (ndirectories eq 0) then begin
    print, 'No directories found, returning!'
    return, ''
  endif

  ;remember all of the subdirectories
```

```
  not_searched = list()
  found = list()
  for i=0, ndirectories-1 do begin
    not_searched.add, dir + path_sep() + directories[i]
    found.add, dir + path_sep() + directories[i]
  endfor

  ;recursivey check for new directories by searching the directories that
  ;we haven't searched
  ;after we search the directory, forget it and only remember
  ;the directories that we ahve found
  while (n_elements(not_searched) gt 0) do begin
    cd, not_searched[0]
    ;search for subdirectories
    subdirs = file_search(/TEST_DIRECTORY, COUNT=ndirectories)
    if (ndirectories gt 0) then begin
      for i=0, ndirectories-1 do begin
        not_searched.add, not_searched[0] + path_sep() + subdirs[i]
        found.add, not_searched[0] + path_sep() + subdirs[i]
      endfor
    endif

    ;remove not_searched directory from the list
    not_searched.remove, 0
  endwhile

  ;convert list to array
  found = found.toarray()

  ;join strings with the path separator for your OS (i.e. ':' or ';')
  found = strjoin(found, path_sep(/search_path))

  ;return to first directory
  cd, first_dir

  ;return the list
  return, found
end
```