Forgive the sin of continuously replying to my own post, but here is the workaround I've used. Spawning the 'dir' command takes much less time than file_search on network shares with many files:

IDL> tic & spawn, 'dir U:\somenetworkshare* /b /aD',found,/hide & toc
% Time elapsed: 0.32800007 seconds.
IDL> tic & found=file_search('U:\somenetworkshare*',count=count,/nosor t) & toc
% Time elapsed: 26.066000 seconds.

So, I wrote a wrapper for the file_search function that determines if the Windows OS is in use, and if the VM mode is not in use, and then does the following:

```
function listfiles,path,pattern,count=count,_extra=e
  if n_elements(pattern) eq 0 then pattern='*'
  if LMGR(/VM) then begin
    return,file_search(path+pattern,/test_regular,count=count,_e xtra=e)   ;forced to use slow version in VM mode.
  endif
  case strupcase(!version.os_family) of
   'WINDOWS':begin
     spawn, 'dir '+path+pattern+' /b /a-D /ON',result,/hide
     count = (result[0] eq '') ? 0 : n_elements(result)
     return,file_dirname(path+pattern,/mark)+result
     end
   'UNIX':begin
      return,file_search(path+pattern,/test_regular,count=count,_e xtra=e)
     end
  endcase
end
```

P.S., I've also found that file_search is slow to return a large list of file matches (>~5000) from a given directory, but not when returning a short list of matches from the same directory. For example, in a directory of ~7000 files, file_search(path+'*') takes around 26 seconds, but file_search(path+'*.txt') returns in 0.3 seconds if there are only a few .txt files.
So, the above workaround actually costs a bit more time for cases where only small file lists are expected.

On Thursday, July 25, 2013 6:44:55 PM UTC-6, b_...@hotmail.com wrote:
> I running IDL 8.2.3 on Win7 64bit. I have an older program that uses findfile() to recursively find a set of filenames with a wildcard. I realize findfile is obsolete, but it runs *much* faster than file_search. When the findfile returns more than ~5000 files, however, I get the following error:
>

>
>
> found=file_search(uval.path+'*',count=count,/mark_dir)
>
> % Array has a corrupted descriptor: FOUND.
>
>
>
> Any ideas what is causing the error?
>
> Assuming that this is a bug that will not be fixed, does anyone have a fast alternative to file_search?
>
>
>
> Thanks