Subject: Re: FILE_SEARCH() issue.
Posted by R.G. Stockwell on Fri, 04 Jun 2004 19:36:25 GMT
View Forum Message <> Reply to Message

"Jonathan Joseph" <jj21@cornell.edu> wrote in message
news:c9qeps$7v0$1@news01.cit.cornell.edu...
> Hello.
>
> I'm seeing some intermittent problems with file_search() not returning
> the full list of files.
...
> What happens is that on rare occasion (though frequently enough to
> cause problems), file_search() does not return the complete list of
> files.
...

I KNOW! :)
I ran into this same problem before, under the same circumstances as you.
It is especially annoying because, in my case, each file was a satellite track, so
my analysis would sneakily drop the occasional orbit. argh!
(only seemed to be a problem when there were ~10,000 files)
I ended up just spawning out the the OS and getting a directory listing.
(and writing a check to make sure that if an orbit was missing, that it
was really missing).

cheers,
bob

Subject: Re: FILE_SEARCH() issue.
Posted by Jonathan Joseph on Fri, 04 Jun 2004 20:26:39 GMT
View Forum Message <> Reply to Message

One of the reasons FILE_SEARCH is so nice though, is that it doesn't run
up agains the OS limit on argument list length. If I try to do an ls
with a wildcard (like *.txt) and specify the full path, I get
"/usr/bin/ls: Arg list too long." Currently, if I actually cd to the
directory, I won't get that particular error, but as files continue to
accumulate, that may not be true for much longer.

It is nice to know I'm not the only one who has seen this problem.

-Jonathan

R.G. Stockwell wrote:

> I KNOW! :)

> I ran into this same problem before, under the same circumstances as you.
> It is especially annoying because, in my case, each file was a satellite track, so
> my analysis would sneakily drop the occasional orbit. argh!
> (only seemed to be a problem when there were ~10,000 files)
> I ended up just spawning out the the OS and getting a directory listing.
> (and writing a check to make sure that if an orbit was missing, that it
> was really missing).
>
> cheers,
> bob
>
>

Subject: Re: FILE_SEARCH() issue.
Posted by R.G. Stockwell on Fri, 04 Jun 2004 21:03:35 GMT
View Forum Message <> Reply to Message

"Jonathan Joseph" <jj21@cornell.edu> wrote in message
news:c9qlvd$b8q$1@news01.cit.cornell.edu...
>
> One of the reasons FILE_SEARCH is so nice though, is that it doesn't run
> up agains the OS limit on argument list length. If I try to do an ls
> with a wildcard (like *.txt) and specify the full path, I get
> "/usr/bin/ls: Arg list too long." Currently, if I actually cd to the
> directory, I won't get that particular error, but as files continue to
> accumulate, that may not be true for much longer.
>
> It is nice to know I'm not the only one who has seen this problem.
>
> -Jonathan

I went to look up exactly what the problem was back when I
ran into the problem with the failure to report all files in a directory.
Turns out that we decided that it was a bug in nsf under redhat 7.0.
We solved it by upgrading to redhat 7.2
So I don't think that helps you since you are using Solaris 8. Sorry.

Cheers,
bob

Subject: Re: FILE_SEARCH() issue.
Posted by K. Bowman on Fri, 04 Jun 2004 23:00:51 GMT
View Forum Message <> Reply to Message

In article <c9qlvd$b8q$1@news01.cit.cornell.edu>,

Jonathan Joseph <jj21@cornell.edu> wrote:

> One of the reasons FILE_SEARCH is so nice though, is that it doesn't run
> up agains the OS limit on argument list length. If I try to do an ls
> with a wildcard (like *.txt) and specify the full path, I get
> "/usr/bin/ls: Arg list too long." Currently, if I actually cd to the
> directory, I won't get that particular error, but as files continue to
> accumulate, that may not be true for much longer.

I think this is a shell (not an OS) limitation. I believe the solution
(under unix systems), is to use find instead of ls.

Ken Bowman

Subject: Re: FILE_SEARCH() issue.
Posted by markcain on Mon, 07 Jun 2004 13:25:20 GMT

View Forum Message <> Reply to Message

FILE_SEARCH does have a bug when searching through subdirectories with folder names containing brackets (i.e \data\bad[folder]here\not_found.txt). All contents below the bracketed file would not be found.

I am using IDL 6.0 on windows. RSI confirmed the problem and it is set to be corrected in future releases. Don't know if problem was on other platform.

It appears that FILE_SEARCH was treating subdirectory names as though they required wildcard expansion (my prognosis).

Hope this helps,
Mark

Kenneth Bowman <k-bowman@null.tamu.edu> wrote in message
news:<k-bowman-FF0F92.17005104062004@news.tamu.edu>...
> In article <c9qlvd$b8q$1@news01.cit.cornell.edu>,
> Jonathan Joseph <jj21@cornell.edu> wrote:
>
>> One of the reasons FILE_SEARCH is so nice though, is that it doesn't run
>> up agains the OS limit on argument list length. If I try to do an ls
>> with a wildcard (like *.txt) and specify the full path, I get
>> "/usr/bin/ls: Arg list too long." Currently, if I actually cd to the
>> directory, I won't get that particular error, but as files continue to
>> accumulate, that may not be true for much longer.
>
> I think this is a shell (not an OS) limitation. I believe the solution
> (under unix systems), is to use find instead of ls.

>
> Ken Bowman

Subject: Re: FILE_SEARCH() issue.
Posted by Paul Van Delst[1] on Mon, 07 Jun 2004 18:52:18 GMT
View Forum Message <> Reply to Message

Kenneth Bowman wrote:
> In article <c9qlvd$b8q$1@news01.cit.cornell.edu>,
>   Jonathan Joseph <jj21@cornell.edu> wrote:
>
>
>> One of the reasons FILE_SEARCH is so nice though, is that it doesn't run
>> up agains the OS limit on argument list length. If I try to do an ls
>> with a wildcard (like *.txt) and specify the full path, I get
>> "/usr/bin/ls: Arg list too long." Currently, if I actually cd to the
>> directory, I won't get that particular error, but as files continue to
>> accumulate, that may not be true for much longer.
>
>
> I think this is a shell (not an OS) limitation. I believe the solution
> (under unix systems), is to use find instead of ls.

I've had the same problem. Liam Gumley pointed me to a solution he uses to remove all the files in the given directory tree which have not been modified in 7 days and whose names begin with AM1:

```
find /modisnfs1/ftp/pub/daac -mtime +7 -name "AM1*" -exec /bin/rm -f {} \;
```

I've adapted this to to wot I needed. Works well.

paulv