
Subject: Re: file_lines and expand path

Posted by [Jean H.](#) on Fri, 04 Aug 2006 18:57:30 GMT

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

Paul van Delst wrote:

> FL wrote:

>

>> Hi,

>>

>> I am implementing file_lines in FL and was wondering what to return
>> when the input path expands to multiple files. The IDL reference guide
>> says nothing. I have run IDL and got curious result:

>>

>> IDL> help, file_lines('*.*.pro')

>> <Expression> LONG64 = 15

>>

>> I have 448 .pro files in the current directory, with a total of 10k
>> lines.

>> IDL seems to be randomly picking a single file and reporting its
>> length (probably the first match in readdir()).

>>

>> What do you expect to get?

>>

>> Possibilities:

>>

- >> 1. never expand (but why do we have NOEXPAND then?)
- >> 2. give an error message if the expansion results in more than one file
- >> 3. return the total length of files
- >> 4. return the total length of files if a new keyword TOTAL was set
- >> 5. do as IDL, pick a single file silently

>

>

> Well, if the routine allows for wildcard inputs, then I wouldn't expect
> *any* of the above.

>

> What I would expect would be an array (in this case 448 elements in
> size) containing the lengths of *each* file that matched the wildcard
> expression, e.g.

what about a structure containing the array you said, and another one
containing the complete file name, so then it becomes fairly easy to
open the files (and the order in which the files are read in the
directory is not a problem). This would allow you to read only files
that have more than X lines in a very convenient way!

Jean

>

```
> IDL> help, file_lines('*.pro')
> <Expression>  LONG64  = Array[448]
>
> Just because IDL does it wrong, doesn't mean you have to. :o)
>
> paulv
>
> p.s. Maybe FL should have the ability to run just like IDL. I.e. you set
> some sort of environment variable, say, FL_RUNLIKE_IDL=true, and it
> performs like IDL (i.e. giving the incorrect result (IMO) as in the
> above example). Set it to false, and FL behaves like a regular person
> would intuitively expect. :o)
>
```
