
Subject: Re: file_basename(): bug or feature?

Posted by [David Fanning](#) on Thu, 24 Feb 2005 21:28:10 GMT

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

Benjamin Hornberger writes:

> On Unix machines, IDL's file_basename() function doesn't split a path in
> Windows format (with backslashes). On Windows machines, it splits both
> (yes, I know, on Windows you can also use both to specify a path). Is
> this a bug or a feature?
>
> Linux:
>
> IDL> filename='d:\path\to\file.ex
> IDL> print,file_basename(filename)
> d:\path\to\file.ex
> IDL> filename='d:/path/to/file.ex'
> IDL> print,file_basename(filename)
> file.ex
>
> Windows:
>
> IDL> filename='d:\path\to\file.ex'
> IDL> print,file_basename(filename)
> file.ex
> IDL> filename='d:/path/to/file.ex'
> IDL> print,file_basename(filename)
> file.ex
>
> I have a situation where a Linux machine has to handle Windows paths,
> where this is quite annoying ...
>
> Any comments?

I'm not sure it's a bug OR a feature. More likely the sensible way to write the program, given the original purpose. :-)

My FSC_BASE_FILENAME program has the same problem, simply because I use PATH_SEP to obtain the path separator. I think it would be difficult to write a generic routine to handle all types of path separators properly, but you could probably write a pretty good specific one if you were sure of the types of filenames you would be getting. I guess FSC_BASE_FILENAME is as good a place as any to start.

http://www.dfanning.com/programs/fsc_base_filename.pro

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Subject: Re: file_basename(): bug or feature?

Posted by [Benjamin Hornberger](#) on Thu, 24 Feb 2005 21:34:06 GMT

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

David Fanning wrote:

- > I think it would be difficult to write
- > a generic routine to handle all types of path separators
- > properly,

Well, with regular expressions it shouldn't be a big deal. But for the time being I just convert all Windows paths to forward slashes ...

Benjamin

Subject: Re: file_basename(): bug or feature?

Posted by [Paul Van Delst\[1\]](#) on Thu, 24 Feb 2005 21:40:53 GMT

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

David Fanning wrote:

- > Benjamin Hornberger writes:

- >
- >

- >> On Unix machines, IDL's file_basename() function doesn't split a path in
- >> Windows format (with backslashes). On Windows machines, it splits both
- >> (yes, I know, on Windows you can also use both to specify a path). Is
- >> this a bug or a feature?

- >>

- >> Linux:

- >>

- >> IDL> filename='d:\path\to\file.ex'

- >> IDL> print,file_basename(filename)

- >> d:\path\to\file.ex

- >> IDL> filename='d:/path/to/file.ex'

- >> IDL> print,file_basename(filename)

- >> file.ex

```
>>
>> Windows:
>>
>> IDL> filename='d:\path\to\file.ex'
>> IDL> print,file_basename(filename)
>> file.ex
>> IDL> filename='d:/path/to/file.ex'
>> IDL> print,file_basename(filename)
>> file.ex
>>
>> I have a situation where a Linux machine has to handle Windows paths,
>> where this is quite annoying ...
>>
>> Any comments?
>
>
> I'm not sure it's a bug OR a feature. More likely the
> sensible way to write the program, given the original
> purpose. :-)
>
> My FSC_BASE_FILENAME program has the same problem,
> simply because I use PATH_SEP to obtain the path
> separator. I think it would be difficult to write
> a generic routine to handle all types of path separators
> properly,
```

Couldn't you just loop over the types of path separators? (e.g. ['/', '\', ':', ...etc] for the various systems? Are any of the path separators on different systems allowable filename characters on others?

paulv

--

Paul van Delst
CIMSS @ NOAA/NCEP/EMC

Subject: Re: file_basename(): bug or feature?
Posted by [James Kuyper](#) on Thu, 24 Feb 2005 21:48:19 GMT
[View Forum Message](#) <> [Reply to Message](#)

Paul Van Delst wrote:

...

```
> Couldn't you just loop over the types of path separators? (e.g.
> ['/', '\', ':', ...etc] for the various systems? Are any of the path
> separators on different systems allowable filename characters on others?
```

Yes, certainly. That's what makes it such a pain. In particular, "\" and

":" are allowable filename characters on Unix systems. It's a little tricky to create a filename containing a "\"" with shell commands, because the "\"" is interpreted by the shell, unless it's either quoted or escaped. Try:

```
echo hello > \  
vi "\"
```

Subject: Re: file_basename(): bug or feature?

Posted by [Paul Van Delst\[1\]](#) on Thu, 24 Feb 2005 22:08:39 GMT

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

James Kuyper wrote:

> Paul Van Delst wrote:

> ...

>

>> Couldn't you just loop over the types of path separators? (e.g.

>> [/, \, :, ..etc] for the various systems? Are any of the path

>> separators on different systems allowable filename characters on others?

>

>

> Yes, certainly. That's what makes it such a pain. In particular, "\"" and

> ":" are allowable filename characters on Unix systems. It's a little

> tricky to create a filename containing a "\"" with shell commands,

> because the "\"" is interpreted by the shell, unless it's either quoted

> or escaped. Try:

>

> echo hello > \
> vi "\"

Oof! I see. I couldn't vi the file, but it was created. Wot a pita.

paulv

--

Paul van Delst

CIMSS @ NOAA/NCEP/EMC
