
Subject: Re: FILE_EXPAND_PATH

Posted by [Mark Hadfield](#) on Thu, 03 Jul 2003 20:44:11 GMT

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

"Jonathan Joseph" <jj21@cornell.edu> wrote in message
news:3F0454DF.A3CFEDF7@cornell.edu...

>
> I have IDL 5.5 on Windows and 5.6 on Solaris. I am using the
> FILE_EXPAND_PATH function, but am not getting the behavior I would
> like on the Windows platform. It works well in all cases except if
> I pass the null string ". On the unix version, I get back what I
> expect, which is the current working directory. On the Windows
> platform I get back garbage: a short string of meaningless
> characters.
>
> Does anyone know if this behavior is fixed in 5.6 for windows? Or
> is this function undefined for the null string?

I hope I'm not violating the beta agreement too badly by saying this, but
no, this behaviour has not changed in IDL 5.6 (beta) on Windows.

FILE_EXPAND_PATH calls on functions in the underlying OS, so it's not too
surprising that its behaviour is platform-dependent in an end case like
this. The function is probably passing an empty, null-terminated string (ie
a single NUL) to the OS's C library and returning whatever comes back. Is
NUL a valid file name on Unix? (Unix as an OS is very permissive about file
names, though Unix applications are less so. I am not convinced that this is
a sensible way of doing things, but let's not go there.)

You should report this to RSI. They may well choose to make the function's
behaviour on Windows more predictable.

--

Mark Hadfield "Ka puwaha te tai nei, Hoesa tatou"
m.hadfield@niwa.co.nz
National Institute for Water and Atmospheric Research (NIWA)

Subject: Re: FILE_EXPAND_PATH

Posted by [Jonathan Joseph](#) on Thu, 03 Jul 2003 21:14:49 GMT

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

Yes, it looks like there is a bug. I received this reply from an
"informed source at RSI" who wished to remain anonymous.

-Jonathan

> Your posting about FILE_EXPAND_PATH caught my attention, so I

- > had a look. You've uncovered a bug in Windows IDL that is really
- > a bug in the underlying Windows GetFullPathName() function...

- > Well...maybe it's an undefined behavior instead of a bug...
- > the Windows documentation is not all that clear on the subject.
- > However, the design/documentation of that function is such that it
- > does not seem that they intended to support it.

- > In any event, I agree with you that IDL's FILE_EXPAND_PATH should
- > return the CWD in this case, as it does for Unix. There are a couple
- > different possible results that might be considered reasonable in
- > this case. One is to return a NULL string, the other is to return
- > the CWD. Of the two, the second seems most useful (though I must
- > say that both seem useless --- this is really a "Don't Care" case).

- > I've added a workaround to IDL for Windows, in which it feeds "."
- > to GetFullPathName() instead of "" in the case of NULL paths,
- > and so, now my development build of IDL gets it right on Windows
- > too. I've checked in a regression test to ensure that it stays
- > fixed.

- > And, I'll see to it that the documentation is updated to say what
- > happens.

- > Now for the bad news... IDL 6.0 is done. It's tested, packaged,
- > and going to hit the duplicator imminently. It would be very, very
- > expensive to stop that process, fix this very minor issue, and
- > then re-roll. So, IDL 6.0 will not have the fix I put in today.
- > The next release following it will...

- > Below, please find file_expand_path2.pro, which you can add to
- > your local IDL library and use as a workaround to your issue until
- > the next IDL shows up. I hope it helps.

- > Thanks...

function file_expand_path2, path

```

; Work around a problem with the real FILE_EXPAND_PATH under MS
; Windows: Passing in a NULL path returns a garbage character
; instead of returning the current working directory. The problem
; stems from the underlying Windows GetFullPathName() function.
;
; This problem exists in IDL 5.4-6.0, but should be fixed in the
; next IDL release after 6.0.
```

```
if (path eq "") then CD, CURRENT=r else r = FILE_EXPAND_PATH(path)
return, r
```

end

Jonathan Joseph wrote:

>
> I have IDL 5.5 on Windows and 5.6 on Solaris. I am using the
> FILE_EXPAND_PATH function, but am not getting the behavior I would
> like on the Windows platform. It works well in all cases except if
> I pass the null string "". On the unix version, I get back what I
> expect, which is the current working directory. On the Windows
> platform I get back garbage: a short string of meaningless
> characters.
>
> Does anyone know if this behavior is fixed in 5.6 for windows? Or
> is this function undefined for the null string?
>
> Thanks.
>
> -Jonathan

Subject: Re: FILE_EXPAND_PATH

Posted by [R.Bauer](#) on Fri, 04 Jul 2003 06:40:36 GMT

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

Jonathan Joseph wrote:

> I have IDL 5.5 on Windows and 5.6 on Solaris. I am using the
> FILE_EXPAND_PATH function, but am not getting the behavior I would
> like on the Windows platform. It works well in all cases except if
> I pass the null string "". On the unix version, I get back what I
> expect, which is the current working directory. On the Windows
> platform I get back garbage: a short string of meaningless
> characters.
>
> Does anyone know if this behavior is fixed in 5.6 for windows? Or
> is this function undefined for the null string?
>
> Thanks.
>
> -Jonathan

Dear Jonathan,

FILE_EXPAND_PATH('.') is on all unix and modern windows the current working directory.

regards
Reimar

--

Reimar Bauer

Institut fuer Stratosphaerische Chemie (ICG-I)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de

a IDL library at ForschungsZentrum Juelich

http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_lib_intro.html

=====