
Subject: Re: Locating a (bitmap) file

Posted by [Dave Greenwood](#) on Tue, 28 Nov 2000 08:00:00 GMT

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

I wrote:

> Using IDL 5.3 and Windows NT I've created an IDL windows application
> (thanks to the kind folks who responded to my recent posting on the
> subject). My application uses a couple of color bitmap labels which
> are stored as .bmp files. The application will be used on multiple
> systems, possibly in a separate location on each system. I'd prefer
> not to hard code the location of the .bmp files. The question is how?

>
[snip]

> I wouldn't mind requiring the .bmp and .sav files to be in the same
> directory if I could somehow find out from within my application
> where the .sav file is located. Is that possible? (I tried !DIR but
> it points to some place in the RSI distribution.)

[snip]

From one of those anonymous RSI/Kodak lurkers came the following suggestion which will solve my problem:

> If you make a call to HELP, CALLS = calls, the return is a
> string array whose first element (calls[0]) contains the name of
> the current routine (which you'll have to STRSPLIT to get out the
> routine name by itself.)

>
> HELP, CALLS = calls
> thisroutine = (STRSPLIT(calls[0], ' ', /EXTRACT))[0]

>
> With that you can call ROUTINE_INFO(name, /SOURCE) to get the
> path to the .sav or .pro file that owns that routine. Find
> the right-most directory separator character for your
> platform (STRPOS(/REVERSE_SEARCH)) in the info.path field,
> then extract the string up to that point.

>
> source = ROUTINE_INFO(thisroutine, /SOURCE)
> CASE STRUPCASE(!version.os_family) of
> 'WINDOWS' : dirsep = '\'
> 'UNIX' : dirsep = '/'
> 'MACOS' : dirsep = ':'
> 'VMS' : dirsep = ']'
> ELSE : dirsep = "
> ENDCASE
> root = STRMID(source.path, 0, STRPOS(source.path, dirsep, /REVERSE_SEARCH))
> Let's say your bitmaps are in a subdirectory named "bitmaps" beneath
> your source directory. You can build the appropriate file name
> via

>
> file = FILEPATH('mybitmapfile.bmp', Root = root, SubDir = ['bitmaps'])
>
> One advantage of using this method is that there is only one step
> that actually requires platform-specific code, where you use
> perhaps a CASE statement based on !version.os_family to define the
> directory separator character.
>
> A second advantage is that you never have to worry about the installation
> directory that your "customer" has defined; directory and file paths are
> based on locations relative to wherever it is they put the .pro/.sav files.

This second advantage is exactly what I wanted to achieve.

Dave

Dave Greenwood Email: Greenwoodde@ORNL.GOV
Oak Ridge National Lab %STD-W-DISCLAIMER, I only speak for myself

Subject: Re: Locating a (bitmap) file
Posted by [Paul van Delst](#) on Tue, 28 Nov 2000 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dave Greenwood wrote:

>
> Using IDL 5.3 and Windows NT I've created an IDL windows application
> (thanks to the kind folks who responded to my recent posting on the
> subject). My application uses a couple of color bitmap labels which
> are stored as .bmp files. The application will be used on multiple
> systems, possibly in a separate location on each system. I'd prefer
> not to hard code the location of the .bmp files. The question is how?
>
> I suppose I could use findfile to search candidate locations, but
> that still requires advance knowledge of each system where the
> application is used.
>
> I wouldn't mind requiring the .bmp and .sav files to be in the same
> directory if I could somehow find out from within my application
> where the .sav file is located. Is that possible? (I tried !DIR but
> it points to some place in the RSI distribution.)
>
> Is there a way to return a color bitmap from an IDL function like you
> can with a b/w bitmap? That way I could compile the function and
> store it in the save file.
>
> Since I'm not particularly experienced with IDL, I'm hoping that I've
> missed some (potentially obvious) way of dealing with this.

Assuming that your IDL directory (say /usr/idl_app) is in the IDL path, you could do something like (on a unix system at least):

```
pos = STRPOS( !PATH, '/usr/idl_app/bmp_datafiles' )

begin_pos = RSTRPOS( !PATH, ':', pos ) + 1
end_pos = STRPOS( !PATH, ':', pos ) - 1
file_path = STRMID( !PATH, begin_pos, end_pos - begin_pos + 1 ) + '/'

file = file_path + 'no1.bmp'
```

This does require the initial hardcoding of the location, but if it's in a generically named directory, I don't see a problem with that (like most things in unix default to /usr/bin or /usr/local/bin etc..).

Alternatively, you could stick your code and data in a subdirectory of the IDL distribution (like the old /user_contrib directory). Either way it will be a bit messy since you have to take into account the different directory delimiters of different systems, mac unix, windoze, vms...oops that's right, no more vms soon. :o(

paulv

--

Paul van Delst Ph: (301) 763-8000 x7274
CIMSS @ NOAA/NCEP Fax: (301) 763-8545
Rm.207, 5200 Auth Rd. Email: pvandelst@ncep.noaa.gov
Camp Springs MD 20746

Subject: Re: Locating a (bitmap) file
Posted by [Dave Greenwood](#) on Wed, 29 Nov 2000 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

"J.D. Smith" <jdsmith@astro.cornell.edu> wrote:

[big snip]

- >
- > Of course, we *could* have hoped RSI would have simply allowed embedded
- > representations of color bitmaps, just as they did black and white
- > ones. Like:
- >
- > widget_button,value=mybmp
- >
- > instead of
- >
- > widget_button,value='my.bmp'
- >
- > Maybe they thought we were not smart enough to encode color icons.

> 5.4.1 perhaps?

I've been given to understand that during the initial design of this feature the engineers queried some folks about which method was preferred. Although the vote was strongly in favor of the former method, obviously it was the latter one that showed up. I also understand that there's at least one feature request in for color icons in arrays.

Dave

Dave Greenwood Email: Greenwoodde@ORNL.GOV
Oak Ridge National Lab %STD-W-DISCLAIMER, I only speak for myself

Subject: Re: Locating a (bitmap) file
Posted by [John-David T. Smith](#) on Wed, 29 Nov 2000 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dave Greenwood wrote:

>

> I wrote:

>> Using IDL 5.3 and Windows NT I've created an IDL windows application
>> (thanks to the kind folks who responded to my recent posting on the
>> subject). My application uses a couple of color bitmap labels which
>> are stored as .bmp files. The application will be used on multiple
>> systems, possibly in a separate location on each system. I'd prefer
>> not to hard code the location of the .bmp files. The question is how?
>>

> [snip]

>> I wouldn't mind requiring the .bmp and .sav files to be in the same
>> directory if I could somehow find out from within my application
>> where the .sav file is located. Is that possible? (I tried !DIR but
>> it points to some place in the RSI distribution.)

> [snip]

>

> From one of those anonymous RSI/Kodak lurkers came the following suggestion
> which will solve my problem:

>

>> If you make a call to HELP, CALLS = calls, the return is a
>> string array whose first element (calls[0]) contains the name of
>> the current routine (which you'll have to STRSPLIT to get out the
>> routine name by itself.)

>>

>> HELP, CALLS = calls
>> thisroutine = (STRSPLIT(calls[0], ' ', /EXTRACT))[0]

>>

>> With that you can call ROUTINE_INFO(name, /SOURCE) to get the

```

>> path to the .sav or .pro file that owns that routine. Find
>> the right-most directory separator character for your
>> platform (STRPOS(/REVERSE_SEARCH)) in the info.path field,
>> then extract the string up to that point.
>>
>> source = ROUTINE_INFO(thisroutine, /SOURCE)
>> CASE STRUPCASE(!version.os_family) of
>>     'WINDOWS' : dirsep = '\'
>>     'UNIX' : dirsep = '/'
>>     'MACOS' : dirsep = ':'
>>     'VMS' : dirsep = '|'
>>     ELSE : dirsep = ""
>> ENDCASE
>> root = STRMID(source.path, 0, STRPOS(source.path, dirsep, /REVERSE_SEARCH))
>> Let's say your bitmaps are in a subdirectory named "bitmaps" beneath
>> your source directory. You can build the appropriate file name
>> via
>>
>> file = FILEPATH('mybitmapfile.bmp', Root = root, SubDir = ['bitmaps'])
>>
>> One advantage of using this method is that there is only one step
>> that actually requires platform-specific code, where you use
>> perhaps a CASE statement based on !version.os_family to define the
>> directory separator character.
>>
>> A second advantage is that you never have to worry about the installation
>> directory that your "customer" has defined; directory and file paths are
>> based on locations relative to wherever it is they put the .pro/.sav files.
>
> This second advantage is exactly what I wanted to achieve.

```

Of course, we *could* have hoped RSI would have simply allowed embedded representations of color bitmaps, just as they did black and white ones. Like:

```
widget_button,value=mybmp
```

instead of

```
widget_button,value='my.bmp'
```

Maybe they thought we were not smart enough to encode color icons.
5.4.1 perhaps?

JD

--

J.D. Smith | WORK: (607) 255-6263

Cornell Dept. of Astronomy | (607) 255-5842
304 Space Sciences Bldg. | FAX: (607) 255-5875
Ithaca, NY 14853 |
