Subject: Re: ERROR_MOD_NOT_FOUND when using call_external
Posted by Brian on Wed, 10 Dec 2003 09:17:31 GMT
View Forum Message <> Reply to Message

Ok, I did what you did and it worked. But now the question is, do I need to write a DLM wrapper to truly use this? In the manual it says

The basic usage of FFTW to compute a one-dimensional DFT of size N is simple, and it
typically looks something like this code:
#include <fftw3.h>
...
{
fftw_complex *in, *out;
fftw_plan p;
...
in = fftw_malloc(sizeof(fftw_complex) * N);
out = fftw_malloc(sizeof(fftw_complex) * N);
p = fftw_plan_dft_1d(N, in, out, FFTW_FORWARD, FFTW_ESTIMATE);
...
fftw_execute(p); /* repeat as needed */
...
fftw_destroy_plan(p);
fftw_free(in); fftw_free(out);
}
(When you compile, you must also link with the fftw3 library,
e.g. -lfftw3 -lm on
Unix systems.)
First you allocate the input and output arrays. You can allocate them in any way that
you like, but we recommend using fftw_malloc, which behaves like malloc except that it
properly aligns the array when SIMD instructions (such as SSE and Altivec) are available
(see Section 3.1.1 [SIMD alignment and
tw malloc], page 15).

I am just confused how I pass an array in IDL to this dll.

-brian

"Brian" <brian.huether@NOdlrSPAM.de> schrieb im Newsbeitrag
news:33269b572f090788e59fd6a07c85fc8d@news.teranews.com...
> Wow. That is interesting! All I did was go to preferences and and add the
> path. I know the path is ok, because I also have a procedure file there and
> IDL finds that procedure ok. Well, I feel like I am a step closer. Can you
> give me more info about your system? What version of IDL, etc?

>
> thanks!
>
> -brian
>
> "Karl Schultz" <kschultz_no_spam@rsinc.com> schrieb im Newsbeitrag
> news:vtbusmdr32aled@corp.supernews.com...
>>
>> "Brian" <brian.huether@NOdlrSPAM.de> wrote in message
>> news:e648955426f3d43b5606a3f898bc6d98@news.teranews.com...
>>> I got the fftw binary from www.fftw.org and am trying to call it from
> IDL.
>> I
>>> have searched the forum and have found little info about this error.
>>>
>>> Here is how I am calling it:
>>>
>>> array = call_external('fftw3.dll', 'fftw_malloc', 32, /CDECL)
>>>
>>> My path is set up so that IDL knows where the dll is. I also get the
> error
>>> without /CDECL. I have also tried prepending the function name with an
>>> undeline as someone in another post that I came across had suggested.
>>>
>>> At this point I have no clue what to do. I don't have the means nor
the
>>> skills to recompile dlls such that they work properly with IDL. I am
>> hoping
>>> this works out of the box.
>>>
>>> Any ideas?
>>
>> I downloaded the binary DLL file, put it in my bin/bin.x86 directory and
> the
>> call_external invocation you listed above works - array gets set to a
LONG
>> value (a pointer, I guess).
>>
>> So, I'm thinking that the problem lies in telling IDL where the dll is.
> If
>> you tell us how you are trying to specify the location of the DLL, maybe
> we
>> can help further. Or just drop the DLL into your distribution
bin/bin.x86
>> directory and get on with it if that isn't too distasteful to you
>>
>> Karl
>>

>>
>
>