
Subject: Re: Do I need a DLM Wrapper for this?

Posted by [Karl Schultz](#) on Wed, 10 Dec 2003 16:37:50 GMT

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

"Brian" <brian.huether@NOdlrSPAM.de> wrote in message

news:42248903bfffbe75ba4af3e74473e176@news.teranews.com...

> I am somewhat new to IDL, so you can imagine how confusing I find the notion

> of DLM wrappers...

>

> In any case, I have made several posts about the FFTW3 dll, and I am getting

> closer to being able to get it working, but now I have come across DLMs, and

> am wondering if I need one (and in general wondering under which circumstances I need one).

>

> Do I need to write a DLM wrapper to truly use this fftw3 dll? 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

You should really look into AUTO_GLUE. Start with the CALL_EXTERNAL documentation and find the discussion on auto glue.

The reasons **why** you need auto glue, and why you just can't pass IDL variables to C entry points, is also described there as well as in the general external development documentation.

That being said, I'm not sure that you'll be able to get away with just auto glue, especially if you have to use a special malloc and pass around "plan" data structures. But you can certainly call and use this lib with a hand-coded DLM.

Karl
