

---

Subject: Re: Pros & cons of methods 1) DLM and 2) Call\_External of calling C from IDL

Posted by [Michael Galloy](#) on Tue, 02 Aug 2016 22:25:01 GMT

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

---

On 8/1/16 1:15 PM, neil.a.salmon@gmail.com wrote:

> Michael,  
>  
> thanks for your good points and the links. Does either of these  
> processes have any impact on the speed with which the C runs called  
> from IDL?

I haven't done any tests, but I doubt there is any significant time difference.

> I'm operating IDL 6.0, so i take it your comments are equally valid  
> from this rather old version?

Yes.

> I take it if you have many C routines you'd like to access from IDL  
> it would probably best to invest in learning the DLM method?

A lot of the disadvantages of the DLM method are mitigated when doing many routines. For example, you only write the boilerplate code once, you only learn the API once, etc. So it is definitely *\*more\** worth it if there are many routines. But the big difference is whether you want to use native features like keywords.

By the way, I have some routines in my library to automatically generate IDL DLM bindings for C routines. You specify the C routines in header-like format, such as:

```
void gsl_rng_env_setup();
IDL_PTRINT gsl_rng_alloc(IDL_PTRINT t);
void gsl_rng_free(IDL_PTRINT r);

void gsl_rng_set(IDL_PTRINT r, unsigned long seed);

unsigned long gsl_rng_get(IDL_PTRINT r);
double gsl_rng_uniform(IDL_PTRINT r);
double gsl_rng_uniform_pos(IDL_PTRINT r);
unsigned long gsl_rng_uniform_int(IDL_PTRINT r, unsigned long n);
```

And my library will output the DLM files. This header is constructed by hand with a bit of knowledge of the IDL DLM API. The original routine's prototypes are the following:

```
const gsl_rng_type * gsl_rng_env_setup (void);
gsl_rng *gsl_rng_alloc (const gsl_rng_type * T);
void gsl_rng_free (gsl_rng * r);

void gsl_rng_set (const gsl_rng * r, unsigned long int seed);

INLINE_DECL unsigned long int gsl_rng_get (const gsl_rng * r);
INLINE_DECL double gsl_rng_uniform (const gsl_rng * r);
INLINE_DECL double gsl_rng_uniform_pos (const gsl_rng * r);
INLINE_DECL unsigned long int gsl_rng_uniform_int (const gsl_rng * r,
unsigned long int n);
```

Let me know if you are interested. I can point you to the code and some examples.

Mike

--

Michael Galloy

[www.michaelgalloy.com](http://www.michaelgalloy.com)

Modern IDL: A Guide to IDL Programming (<http://modernidl.idldev.com>)

---