Subject: Re: DLM 32/64 bits
Posted by Abraham campbell on Mon, 19 May 2008 16:16:11 GMT
View Forum Message <> Reply to Message

On May 15, 1:39 am, lecacheux.al...@wanadoo.fr wrote:
> This is maybe a question more related to Windows than to IDL: is there
> a way to use, within IDL, a 32 bits DLM (i.e. 32 bits dll) while
> running a 64 bits system (XP64 or Vista) ? Of course, DLL sources are
> not available for being recompiled in 64 bits.
> Thanks for any insight.
> alx

I'm guessing that you can't mix and match 32- and 64-bit object code in the same process. However, there is an undocumented feature in IDL 7.0 for the idl_idlbridge object that might help.

Normally, when you create an instance of the idl_idlbridge object, it creates an Out-of-Process Server (which loads IDL) in the same bitness as the parent process. For example, a 64-bit parent process creates a 64-bit OPS child process. You can then interact with that OPS/IDL using the idl_idlbrddge interfaces 'execute', 'getvar', 'setvar', etc. In IDL 7.0, there is an undocumented keyword 'ops' that allows you to control the bitness of the OPS/IDL child process that gets created. You can specify either '32' or '64' as the 'ops' keyword value, e.g. "ops=32" or "ops=64". Then, calling 'setvar' and 'getvar' automatically marshals the data correctly between the different bit processes. There is some error checking for such things as trying to marshal a very, very large array from a 64-bit process back to a 32-bit process, where the array is too large to fit into a 32-bit address space.

For example,
To create a 32-bit OPS process:

```
o32 = obj_new('idl_idlbridge',ops=32)
```

To create a 64-bit OPS process:

```
o64 = obj_new('idl_idlbridge',ops=64)
```

You can even mix creating many 32- and 64-bit child processes from the same parent process.

So, you could use this to create an idl_idlbridge process in 32-bit mode from your 64-bit parent process. Then, load/use your 32-bit DLM in that 32-bit OPS/child process. You could then get and set data from the 32-bit child process.

Similarly, if you are in a 32-bit parent process, you can create a 64-bit child process, in order to use 64-bit DLMs that can handle very large data.

This works on all supported platforms, though there are some known issues with trying to use this undocumented keyword in 'run-time distributions' where IDL is not officially installed, but all the files are just copied. But, if you are running with an installed version of IDL, it should work. However, since it's 'undocumented' it works 'as-is' and there is no official tech support for it. So, use at your own discretion.

Abraham

Subject: Re: DLM 32/64 bits
Posted by lecacheux.alain on Tue, 20 May 2008 20:51:58 GMT

View Forum Message <> Reply to Message

On 19 mai, 18:16, abra...@ittvis.com wrote:

> On May 15, 1:39 am, lecacheux.al...@wanadoo.fr wrote:
>
> I'm guessing that you can't mix and match 32- and 64-bit object code
> in the same process. However, there is an undocumented feature in IDL
> 7.0 for the idl_idlbridge object that might help.
>
> Abraham

Thanks for the tip. Since the idl_idlbridge object was already defined in 6.4, would it be possible to use it with this version instead of
7.0 ?
alx.