
Subject: Re: portability of programs written under IDL3.6.1 to IDL5.2

Posted by [thompson](#) on Wed, 10 Mar 1999 08:00:00 GMT

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

David Foster <foster@bial1.ucsd.edu> writes:

> Bart -

> Porting your programs shouldn't be much of a problem, though you
> will have to change your WIDGET_BASE() calls for top-level bases,
> as well as calls to XMANAGER, in your widget programs. RSI introduced
> the /NO_BLOCK keyword which gives you back your command-line after
> registering a widget, and this necessitates the above changes. Also,
> RSI changed the behavior of modal widgets, and moved this /MODAL
> keyword. The changes are minimal, but if you have a large collection
> of widget programs it may take some time to update them. Be aware
> that until you do most of your widget programs won't work.

Is this really true? Although the recommendation now is to put /MODAL in the call to WIDGET_BASE instead of XMANAGER, doing it the old way does still work. We had some problems with modal widgets that called non-modal subwidgets, but RSI gave us a simple fix to xmanager.pro that solved that problem. (It's a good thing, too, because the newly recommended way of implementing modal widgets is simply unacceptably limited, IMHO.) I'd be glad to share that modified xmanager.pro with anybody who wants it.

I'm not saying you shouldn't modify your code to conform to currently accepted practice. I'm just saying that, for most things, previously written software should still work as before, so it may not be as big a job porting the software as it sounds, so long as you don't mind missing out on some of the newer features such as /NO_BLOCK.

Unfortunately, I don't have any experience of using CALL_EXTERNAL in a Windows environment, so I can't comment on the original question about programming calls to DLLs.

William Thompson

=====
=====

> Vekemans Bart wrote:

>>

>> Years ago our Institute purchased IDL3.6.1 and since then
>> I am working on a program that simultaneously can control
>> several hardware cards by means of CALL_EXTERNAL, which
>> invokes functions of the DLL libraries delivered with
>> the various cards. Until now I was using 16-bit DLLs, as
>> this version of IDL is restricted to calls to 16-bit DLLs

>> in Windows.
>> Unfortunately, new cards are now delivered with 32-bit DLLs,
>> and as such I cannot use the delivered libraries of these
>> new cards. I have now the following question. Is my program
>> portable from IDL3.6.1 to IDL5.2 or has the programming
>> environment changed so much that I have to rewrite my
>> whole program when I want to upgrade to IDL5.2 ?
>> I want to try to find someone who experienced the same problems,
>> and I would be very gratefull to her/him if she/he wants
>> to share her/his experience about this problem.
>>
>> Waiting for someone to answer,
>>
>> greetings from Antwerpen (Belgium, Europe),
>>
>> Bart Vekemans
