
Subject: Re: IDL make_dll and unresolved externals
Posted by [Wout De Nolf](#) on Wed, 03 Feb 2010 10:50:01 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Tue, 2 Feb 2010 18:13:34 -0800 (PST), james
<james5452003@yahoo.com> wrote:

> I think my problem currently is with make_dll, and not with
> call_external. After make_dll is called, I have an unresolved
> external error at compile time.
> I do set the header directories and .lib file locations parameters for
> the make_dll command, though. As for finding the opencv dll's, I'm
> not sure how to direct its location using the make_dll command. Is
> there somewhere I can set this on the IDL workbench?
>
> I am unfamiliar with dependency walker and idl_opserver. I will look
> into these. (I am relatively new to IDL)

Ah, sorry. Or to quote Kenneth: RTFQ :-).

So make_dll can't make the dll and gives a very helpful "unresolved external error" message.

The next obvious question: can you compile and build the C-code without make_dll (i.e. C compiler and linker)? Btw, this is C code and not C++ right?

As for the location of the dll. This is only needed when calling your dll, not when building it. You don't need it at all if you statically link to opencv.

Dependency walker checks export functions (and decoration) as well as external dependencies. This helps you in finding out what's wrong when call_external throughs an error. Since you can't even build the dll, this will not help you at this stage :-).

idl_opserver is running when you start IDL. If you have a debugger than you could attach that debugger to idl_opserver, set a breakpoint in your C code and call "call_external" in IDL. It will break and you can trace through your C-code to find out what's wrong. Here again: this only helps in debugging the dll when you call its functions.
