Subject: Re: Windows IDL DLM with Cygwin
Posted by Mark Hadfield on Tue, 12 Jul 2005 22:06:19 GMT
View Forum Message <> Reply to Message

joey@swri.edu wrote:
> Ok, thanks to all who have replied. I believe I got the answer I needed in
> that "using a Cygwin compiled DLL is impossible with IDL".

The techtip reported by Haje

http://www.rsinc.com/services/techtip.asp?ttid=3798

shows that it *is* possible to use Cygwin-compiled DLLs with IDL, via CALL_EXTERNAL. I don't see any mention of the -mno-cygwin switch in the techtip, so it looks like these are true Cygwin DLLs.

Of course DLMs are another matter, but this techtip suggests that maybe you should persist.

A note for those who are completely mystified by this thread:

Cygwin (http://sources.redhat.com/cygwin/) is a Unix-emulation system for Windows. The Cygwin package includes a myriad of different tools, one of which is the GCC compiler system, includes the C compiler, gcc. Normally an executable produced by gcc will require the Cygwin DLL at run time. It is this DLL that provides the Unix API functions. It is also possible to use Cygwin's gcc to produce an executable that doesn't require the Cygwin DLL. The command-line switch to accomplish this is "-mno-cygwin".

Mingw (http://mingw.sourceforge.net/index.shtml) is a port of the Gnu tools to Windows that doesn't rely on a separate run-time DLL. It does not emulate Unix, so porting Unix software to Mingw is harder than porting it to Cygwin. (The latter is often trivial.) An executable produced by Mingw's gcc will look very much like one produced by Cygwin's gcc with -mno-cygwin. In fact Cygwin bundles its own version of the Mingw headers and import libraries to achieve its "-mno-cygwin" functionality.

--
Mark Hadfield "Kei puwaha te tai nei, Hoea tahi tatou"
m.hadfield@niwa.co.nz
National Institute for Water and Atmospheric Research (NIWA)