Subject: Re: In case someone has trouble including export.h
Posted by James Kuyper on Tue, 29 Jan 2002 19:57:45 GMT
View Forum Message <> Reply to Message

Ruediger Kupper wrote:

> Hi!
>
> We just ran into a problem with IDL5.5's export.h file, when
> including it from a c++ source file:
> I tested g++-2.95 and g++-3.0, and both (most annoyingly)
> choked on it, due to the following reason:
>
> The original export.h shipping with IDL 5.5 featured a prototype
> declaration with a variable named "template". This choice of name
> effectively prevents the code from compiling, when included from C++
> code (even when included 'extern "C" {}').
...

> The IDL_ARG_PROTO(...) macro is defined near the beginning of the file.
> It acts as a null-filter, reproducing it's argument. It can be used to
> completely turn off prototypes in this file, for compatibility to
> non-ANSI-compilers. We could have utilised this by #define-ing
> IDL_CC_NOT_ANSI, which also seemed to fix the above mentioned problem.
> But I felt this being much more interfering than simply respelling the
> name.

I've been exchanging e-mail about this with you, and I think we've
reached a point where I should bring my results back to this forum.

The export.h file was apparantly intended to work with C++, as indicated
by the "#ifdef __cplusplus" lines in it. However, since it contains the
C++ keyword 'template', used as a variable name, it can't actually be
compiled in C++.
Your solution was to compile it by #defining IDL_CC_NOT_ANSI. However,
that removes the arguments from every function prototype declared using
the IDL_ARG_PROTO() macro.

In C, that's legal: missing arguments means that the actual number and
types of the arguments are unspecified. It's up to the programmer to
make sure that the arguments are of the correct type and number.

However, in C++, if a function declaration contains no arguments, that
indicates that the function takes no arguments. It's an error to call
the function with arguments. Therefore, the following dumb example
cannot be compiled with C++, whether or not you #define IDL_CC_NOT_ANSI:

/* This shouldn't be needed. export.h should #include it: */

```
#include <stdio.h>

#include "export.h"

void func(void) { }

int main(void)
{
    IDL_LONG length=0;
    IDL_TIMER_CB callback = &func;
    IDL_TIMER_CONTEXT context=0;
    IDL_TIMER_CONTEXT_PTR pcontext=&context;

    IDL_TimerSet(length, callback, 1, pcontext);
    (*callback)();

    return 0;
}
```