The C runtime library cannot be mixed and matched

September 11, 2015

Raymond Chen

In 2011, a customer had an application written in C++ with Visual Studio 2003 that consumes a static library provided by a third party, let's call it `contoso.lib` . Now, `contoso.lib` is a static library compiled with Visual C++ 6. The customer is migrating from Visual Studio 2003 to Visual Studio 2008, but they are still using that old `contoso.lib` library that was compiled with Visual C++ 6. They were afraid that they would encounter some unresolved externals due to name mangling issues, but they were pleasantly surprised when there were no such issues.

Now the questions.

1. Is it correct to link a VC6 static library into a VS2008 project?
2. Even if the linking is successful, do you see any issues or disadvantages with this approach?

The customer liaison's opinion was "Due to missing security features like SAFESEH, GS, DYNAMICBASE, and NXCOMPAT, there may be a lot of drawbacks to using VC6 libraries in a VS2008 project. What do you think?"

It's nice that you're thinking about the security features added in recent versions of Visual Studio, using a generous definition of *recent* to mean *less than nine years old*. But the issue is more fundamental than just security. The issue is correctness.

You cannot mix libraries across compiler versions. That you're trying to mix libraries with compiler versions that are *nineteen years apart in age* is mind-boggling. The Win32 ABI does not extend into compiler-specific behavior, like its internal lookup tables for exception dispatching, private helper functions for RTTI, class member layout, the order of base classes, the implementations of STL classes, the layout of various internal structures, or what happens if you return `FALSE` from `DllMain`.

Name mangling will not catch any of these issues. If you modify a class, say by adding a new member variable or base class, the mangled name does not change, even though the new class is probably incompatible with the old one.

Assuming you manage to dodge all the link errors, what will happen is that these discrepancies will manifest themselves as random failures or memory corruption *at run time*.

You will have to go back to Contoso and ask them for a version of the library that is compatible with Visual Studio 2008.

(Then again, since this question was asked in 2011, they may want to go straight to Visual Studio 2010, which was the most recent version of Visual Studio available at the time.)

Bonus chatter: Another solution is to create a project in Visual Studio 2003 whose sole job is to wrap the static library in a DLL. The rest of your program can be developed in Visual Studio 2008, using the DLL interface to access the static library.

Raymond Chen

Follow