Understanding errors in classical linking: The delay-load catch-22

devblogs.microsoft.com/oldnewthing/20130111-00

January 11, 2013

Raymond Chen

Wrapping up our week of understanding the classical model for linking, we'll put together all the little pieces we've learned this week to puzzle out a linker problem: The delay-load catch-22.

You do some code cleanup, then rebuild your project, and you get

```
LNK4199: /DELAYLOAD:SHLWAPI ignored; no imports found from SHLWAPI
```

What does this error mean?

It means that you passed a DLL via the `/DELAYLOAD` command line switch which your program doesn't actually use, so the linker is saying, "Um, you said to treat this DLL special, but I don't see that DLL."

"Oh, right," you say to yourself. "I got rid of a call to `HashString`, and that was probably the last remaining function with a dependency on `SHLWAPI.DLL`. The linker is complaining that I asked to delay-load a DLL that I wasn't even loading!"

You fix the problem by deleting `SHLWAPI.DLL` from the `/DELAYLOAD` list, and removing `SHLWAPI.LIB` from the list of import libararies. And then you rebuild, and now you get

```
LNK2019: unresolved external '__imp__HashData' referenced in function 'HashString'
```

"Wait a second, I stopped calling that function. What's going on!"

What's going on is that the `HashString` function got taken along for the ride by another function. The order of operations in the linker is

- Perform classical linking
- Perform nonclassical post-processing
  - Remove unused functions (if requested)
  - Apply `DELAYLOAD` (if requested)

The linker doesn't have a crystal ball and say, "I see that in the future, the 'remove unused functions' step is going to delete this function, so I can throw it away right now during the classical linking phase."

You have a few solutions available to you.

If you can modify the library, you can split the `HashString` function out so that it doesn't come along for the ride.

If you cannot modify the library, then you'll have to use the `/IGNORE` flag to explicitly ignore the warning.

Exercise: Another option is to leave `SHLWAPI.LIB` in the list of import libraries, but remove it from the `DELAYLOAD` list. Why is this a dangerous option? What can you do to make it less dangerous?

Raymond Chen

Follow