Why are Bluetooth functions exported from the infrared control panel?

devblogs.microsoft.com/oldnewthing/20190516-00

May 16, 2019

Raymond Chen

Remember infrared communications?

Okay, maybe you don't. It was a thing for a while in the late 1990's and early 2000's, but it never really caught on, and it became overshadowed by other short-range wireless communication mechanisms like Wi-Fi and Bluetooth.

If you peek inside the the infrared control panel `irprops.cpl`, you'll see that it exports a ton of Bluetooth functions. What is Bluetooth doing in the infrared control panel?

Rewind to 2004. Windows XP Service Pack 2 was under development, and one of the features being added was support for Bluetooth. However, at the time, the service pack rules prevented adding new DLLs to the system. The Bluetooth folks had to squeeze their new APIs into the existing infrared control panel.

The restriction on introducing new binaries to the system was subsequently lifted, but the functions had to remain in `irprops.cpl` for compatibility. Those functions nowadays merely forward to the implementation in the real Bluetooth DLL `bthprops.cpl`.

For over a decade, the documentation has exhorted people to use `bthprops.lib` rather than `irprops.lib`.

And finally, the Windows SDK pulled the plug. It no longer includes a copy of `irprops.lib`, thereby removing the temptation to link with the wrong library. If you want to call those Bluetooth functions, link with `bthprops.lib` already.

Note that if you do this, your program won't run on Windows XP Service Pack 2, because Windows XP Service Pack 2 doesn't have a `bthprops.cpl`. For the hopefully none of you that are developing software to run on Windows XP Service Pack 2 (and really, who are you people?), you can copy `irprops.lib` from your Windows XP SDK and link with that.

Bonus chatter: Why not just ship `irprops.lib` as a duplicate of `bthprops.lib`? That way, people are silently upgraded to the correct library.

Sure, that would silently upgrade people to the correct library, but it also means that if you are one of those hopefully none of you that is writing code that intends to run on Windows XP Service Pack 2, your code silently broke: Your program that used to run on Windows XP Service Pack 2 no longer works, and there was no warning or error message anywhere along the way.

We decided that it was better to make the break obvious.

Raymond Chen

Follow