In the Windows kernel, what is a LUID, and what makes it loo-ey?

devblogs.microsoft.com/oldnewthing/20240830-00

August 30, 2024

Raymond Chen

In the Windows kernel, you will see a thing called a `LUID`, commonly pronounced /loo-id/. The kernel documentation says

> The **LUID** structure is an opaque structure that specifies an identifier that is guaranteed to be unique on the local machine. For more information, see the reference page for **LUID** in the Microsoft Windows SDK documentation.

If you go to the Windows SDK documentation, you get

> Describes a local identifier for an adapter.
>
> **Remarks**
>
> This structure is used by the ID3D12Device::GetAdapterLuid and GetSharedResourceAdapterLuid methods.

Somehow, the display driver folks took over the `LUID` documentation and made it be all about display drivers. It's as if the file system team had taken over the `LARGE_INTEGER` documentation and made it say "The `LARGE_INTEGER` structure holds the size of a file in bytes" because the `GetFileSizeEx` function uses the `LARGE_INTEGER` structure for that purpose.

Really, a `LUID` is a structure that holds a 64-bit integer (broken into two 32-bit parts). The 64-bit integer is "locally unique", in the sense that it will not match any other `LUID` generated from the same system until the system is rebooted.

You can ask for a `LUID` to be generated for you by calling `AllocateLocallyUniqueId`.

Since `LUID`s are only unique to the system, you probably shouldn't send them to other systems (since they won't be unique there). And since `LUID`s lose uniqueness when the system reboots, you probably shouldn't save them anywhere persistent, because they won't make sense after a reboot. The purpose of a `LUID` is to let the system identify things whose lifetimes do not extend beyond a reboot.