

Why not auto convert week-based time zone information to date-based?

devblogs.microsoft.com/oldnewthing/20160304-00

March 4, 2016



Raymond Chen

When I described the two ways that the `TIME_ZONE_INFORMATION` structure represents the rules for daylight saving time, commenter Voo [argued](#) (and I hope I interpreted the message correctly) that either there should be two functions, one for each of the ways, or have `GetTimeZoneInformationForYear` function calculate the day automatically from the week and return that. That way, callers always see the correct day.

Okay, there are multiple things in play here, but they all boil down to “no code is an island”.

If all you needed to support was a way to answer the question “When does daylight saving time start and end in this specific year”, then certainly the *n*th week rules could be interpreted by the `GetTimeZoneInformationForYear` function, so that all that came out of the function was a specific date and time.

But that’s not the only function that deals with time zone information. For starters, we have `GetTimeZoneInformation`. Since this function is not dependent on the year, it wouldn’t know what year to use to calculate the date on which the third Sunday falls. It would have to leave the structure in generic form.

Now, maybe you say, “Yeah, these two functions return the same structure, and you just have to know that one of the functions will autoconvert and the other won’t.” Some people will find that confusing.

Maybe you solve that problem by saying, “These two functions return different structures, reflecting their different natures.” Some people will be frustrated by that, because they will have to write two versions of the same function, one for each structure.

(I’m not even going to bother adding `GetDynamicTimeZoneInformation` to the mix.)

And then there’s the question of whether it’s okay that the `Get` function returns a structure that cannot be passed into the corresponding `Set` function. (Which is a bit of a pointless question here, seeing as there is no `SetTimeZoneInformationForYear` function, but

maybe someday it will show up.) It also means that you cannot compare two `TIME_ZONE_INFORMATION` structures from different years to see whether the time zone transition rules are the same in those years. (This is, admitted, a fringe scenario.)

Two functions seems the less crazy way out. There would be a `GetUnexpandedTimeZoneInformationForYear` which returns the unprocessed information and a `GetExpandedTimeZoneInformationForYear` that does the date lookup for the appropriate year. Though personally, I would keep `GetTimeZoneInformationForYear` as-is and add a `CalculateDaylightSavingTransitionDatesForYear` function. That way, you could use it to expand other things.

I suspect no such `CalculateDaylightSavingTransitionDatesForYear` function exists because the kernel folks have a general bias against writing functions that you could have written yourself, so that they can focus on writing functions that you *can't* write yourself because it requires special access to things that only the kernel has access to. (Like querying the current time zone.)

Raymond Chen

Follow

