Converting from a UTC-based SYSTEMTIME directly to a local-time-based SYSTEMTIME

devblogs.microsoft.com/oldnewthing/20140307-00

March 7, 2014

Raymond Chen

Last year, I presented this commutative diagram

A 2-by-2 grid of boxes. The top row is labeled FILETIME; the bottom row is labeled SYSTEMTIME. The first column is labeled UTC; the second column is labeled Local. The upper left box is labeled GetSystemTimeAsFileTime. There is an outgoing arrow to the right labeled FileTimeToLocalFileTime leading to the box in the second column labeled None. There is an outgoing arrow downward labeled FileTimeToSystemTime leading to the box in the second row, first column, labeled GetSystemTime. From the box in the upper right corner labeled None, there is an outgoing arrow downward labeled FileTimeToSystemTime leading to the box in the second row, second column, labeled GetLocalTime.
UTC
Local
FILETIME
GetSystemTimeAsFileTime
FileTimeToLocalFileTime
(None)
FileTimeToSystemTime
FileTimeToSystemTime
SYSTEMTIME
GetSystemTime
GetLocalTime
→
↓
↓
I claimed that there was no function to complete the commutative diagram by connecting the bottom two boxes. I was wrong, but I'm going to try to get off on a technicality. You can connect the two boxes by calling `SystemTimeToTzSpecificLocalTime` with `NULL` as the time zone parameter, which means "Use the current time zone."

The same diagram as above, but there is a new arrow connecting GetSystemTime to GetLocalTime labeled SystemTimeToTzSpecificLocalTime.
UTC
Local
FILETIME
GetSystemTimeAsFileTime
FileTimeToLocalFileTime
(None)
FileTimeToSystemTime
FileTimeToSystemTime
SYSTEMTIME
GetSystemTime
SystemTimeToTzSpecificLocalTime
GetLocalTime
→
↓
↓
→

This works here because the time being converted always refers to the current time. Here comes the technicality. This technique doesn't work in general because `SystemTimeToTzSpecificLocalTime` uses the time zone in effect *at the time being converted*, whereas the `FileTimeToLocalFileTime` function uses the time zone in effect *right now*. Furthermore, it doesn't take into account changes in daylight savings rules that may have historically been different from the current set of rules. (Though this is easily repaired by switching to `SystemTimeToTzSpecificLocalTimeEx`.) The trick works here because the time we are converting is *right now*. In other words, the more general diagram does not commute. Instead, it looks more like this:

Same as before, but this time the boxes are unlabeled, and the bottom right box is split in two. The inbound arrow from the left goes to one box and the inbound arrow from the top goes to another box. The two halves of the split boxes are marked as not equal.
UTC
Local
FILETIME
FileTimeToLocalFileTime
FileTimeToSystemTime
FileTimeToSystemTime
SYSTEMTIME
SystemTimeToTzSpecificLocalTimeEx
≠
→
↓

↓
→

This is why the documentation for `FileTimeToLocalFileTime` tells you that if you want to get from the upper left corner to the upper right corner while accounting for daylight saving time relative to the time being converted, then you need to take the long way around.

So what we have is not so much a commutative diagram as a something like covering space: If you start at any box and travel around the diagram, you won't necessarily end up where you started. Let's start at the upper left corner for the sake of example.

Back to the four-box diagram, with empty boxes. The arrows follow a clockwise path. From the upper left, we go to the upper right via FileTimeToLocalFileTime, then to the bottom right via FileTimeToSystemTime, then to the bottom left via TzSpecificLocalTimeToSystemTimeEx, then back to the upper left via LocalFileTimeToFileTime.
UTC
Local
FILETIME
FileTimeToLocalFileTime
SystemTimeToFileTime
FileTimeToSystemTime
SYSTEMTIME
TzSpecificLocalTimeToSystemTime
→
↓
↓
→

When you return to the upper left box, you might end up somewhere else, probably an hour ahead of or behind where you started. Each time you take a trip around the diagram, you drift another hour further away. Well, until you hit another daylight saving time changeover point.

Raymond Chen

Follow