

Kick Earth Away from My Calendar

Jim Hacker faced a choice between a bishop who wanted to kick the Queen out of the Church of England and one who wanted to kick God out of it. I propose a similarly radical choice for our timekeeping systems: kick Earth away from my calendar, and kick the calendar away from elapsed-time counting.

The recurring drama over leap seconds illustrates what happens when people confuse three fundamentally different things: the measurement of elapsed time, the representation of dates on a calendar, and the astronomical behavior of a planet. By insisting on maintaining an artificial synchronization between them, we have created unnecessary complexity, operational disruption, and a perpetual obligation to amend software that ought to remain correct indefinitely.

Time is not a calendar, and neither is Earth

An elapsed-time counter should do one thing: count physical seconds from a defined epoch. It should not care whether Earth is spinning faster or slower, whether the Sun appears overhead at noon, or whether a particular civilization has decided to reorganize its months.

A calendar, meanwhile, is a human-designed coordinate system for expressing instants in a form convenient for civil life. The Gregorian calendar represents a date and time using six parameters: year, month, day, hour, minute, and second. These parameters are conventions, not laws of nature.

Rotation of Earth is an entirely different matter. Its angular velocity can increase or decrease at any time. The length of a solar day is not perfectly constant, and no mathematical or physical principle requires the Sun to reach its highest point in the sky at precisely 12:00:00 local clock time every day.

What utility do we gain by forcing these systems into perpetual agreement? The practical benefit of making every astronomical noon correspond exactly to a particular clock reading is negligible. The engineering and operational costs of maintaining that correspondence, however, are real.

Leap seconds are the absurd consequence of treating a variable astronomical phenomenon as a mandatory correction to a stable system of timekeeping. Instead of accepting that clocks measure time while astronomy describes celestial motion, we keep adjusting the former to imitate the latter. The result is a mess for software systems that need a consistent, predictable timeline.

A calendar should be a mathematical function, not a moving target

Once adopted, a calendar should be a fixed, stateless mathematical mapping between two domains:

- A physical epoch-second counter integer (PESCI), measuring elapsed physical seconds from a defined epoch.
- A Cartesian product of calendar parameters, such as the six fields of a Gregorian date and time.

The conversion should work in both directions. Given a PESCI, the calendar produces a date and time. Given a valid date and time, it produces the corresponding PESCI. Both operations should be deterministic, reproducible, and independent of the future behavior of Earth or any other planet.

Consider the date 33550336-11-12 11:22:33. Under a specified calendar and epoch definition, this should represent one particular PESCI. That association must not change because Earth changes its angular velocity, because a future astronomer discovers a new irregularity in planetary rotation, or because someone decides that noon ought to remain astronomically meaningful.

There is one important qualification: the calendar must define its arithmetic precisely, including leap-year rules, valid field ranges, and the relationship between its seconds and the chosen physical time scale. A calendar label cannot identify a unique physical instant without those definitions. Nor should the term physical second be confused with an unspecified clock reading: the epoch and time scale must be explicit.

But once those rules are fixed, they should stay fixed. A calendar is a coordinate system, not a subscription service whose terms can be revised whenever the planet misbehaves.

Choose new calendars instead of modifying calendars

What happens when an established calendar no longer suits the astronomical reality or lifestyle of the society using it? The answer is simple: adopt another calendar.

A calendar designed for a planet with one rotational period may become inconvenient if the rotation of a planet changes significantly. A society might then choose a different arrangement of days, months, and years. It might even decide to migrate to a new calendar every 10,000 years, if that proves useful.

There is nothing inherently wrong with such a migration. Calendar conventions are human choices, and human choices can be revised. The essential requirement is that the new calendar be introduced as a new, explicitly defined mapping, while the old mapping remains available for interpreting historical records.

This is no different in principle from changing a coordinate system, adopting a new measurement convention, or introducing a new file format. A new representation does not require us to corrupt the old one.

Most importantly, the migration must not rewrite the meaning of elapsed time. A physical instant that occurred under the old calendar remains the same physical instant under the new one. Only its human-readable representation changes.

Correct software must not become incorrect by decree

Software implementing a precisely specified calendar should not become incorrect merely because future generations impose additional exceptions on their timekeeping systems. If an implementation conforms to its specification, it remains correct with respect to that specification.

This is a basic principle of dependable engineering. A cryptographic algorithm should not silently change its mathematics because the political climate has changed. A measurement conversion should not acquire a new exception because somebody dislikes the result. And a calendar library should not have its established semantics repeatedly invalidated because civil authorities insist on preserving a particular relationship with astronomical phenomena.

There is, of course, a distinction between a software defect and a change in requirements. Software that fails to implement its stated rules is wrong. Software that faithfully implements a stable specification does not become wrong merely because a new specification is introduced. The latter calls for a new implementation, an explicit configuration, or a separately versioned convention — not a retrospective declaration that previously correct behavior was incorrect.

Continually imposing new exceptions on established systems is irresponsible engineering. It creates avoidable risks, complicates interoperability, and shifts the burden of arbitrary policy decisions onto every developer who depends on reliable timekeeping. The insistence that all systems must accommodate every new exception is not a virtue of flexibility; it is a failure to respect the boundary between a stable technical foundation and a changing human convention.

Bad planet, bad

We should stop asking elapsed time to follow a calendar, and stop asking a calendar to follow the changing rotation of Earth.

Let elapsed-time counters count seconds. Let calendars translate those counters into human-readable dates. Let astronomy describe the actual movements of celestial bodies. Where civil society needs a closer relationship between clock time and the Sun, let it adopt an appropriate convention at the civil-time layer, without rewriting the underlying timeline.

The proposal is not to abolish calendars, disregard astronomy, or forbid societies from changing their conventions. It is to separate responsibilities that never needed to be entangled in the first place.

Earth may change its speed of rotation. Civilizations may change their calendars. Neither should be permitted to change the meaning of a previously defined elapsed-time counter.

Kick Earth away from my calendar. And, while we are at it, kick the calendar away from elapsed-time counting.