# Panel Discussion on Time Rollover Events and Leap Seconds

Judah Levine
Time and Frequency Division
National Institute of Standards and Technology
Boulder, Colorado 80303

*Abstract*

***The panel discussed the rollover in GPS time that will occur in 1999, the rollover in the civil date at the beginning of the year 2000 and the problems associated with leap seconds. We showed why these problems arise and we discussed various strategies for dealing with them. We also outlined difficulties that arise because of a lack of standardization in data formats and signaling standards. Judah Levine of NIST was the moderator; other members of the panel were David Mills, University of Delaware, Don Mitchell from TrueTime and Edward D. Powers from the US Naval Observatory.***

## THE CAUSES OF ROLLOVERS AND LEAP SECONDS

The Global Positioning System time is expressed using two parameters: the week number and the second of the week. Week number 0 began at 0000 UTC on 6 January 1980, and week number 1023 (the largest possible week number) will begin on 15 August 1999. The week number will wrap around to 0 at the start of the following week (22 August) and the cycle will then repeat. The GPS time message does not contain any indication of this rollover, so that all GPS times have an ambiguity of 1024 weeks.

Systems that use only two decimal digits to express the year will face a similar problem starting on 1 January 2000 -- the century associated with a year specified in this way will be ambiguous. In addition to these rollover problems, many databases use the year "99" as a flag meaning "forever." Dates such as 9/9/99 or 11/11/99 are often used for this purpose, since it is easy to enter them and they do not violate the 6-digit format expected by software that processes the date field.

The ambiguities produced by rollovers are a feature of all digital systems -- values are always represented in a finite-length field, and the maximum-possible value will be reached sooner or later. The value will wrap around to 0 after that and start over again, and there is generally no way of indicating this rollover in the time tag itself. The size of the field and, therefore, the interval between rollover events, is determined by a design trade-off: larger fields will have less frequent roll-overs, but the time tags will take up more space and will generally require more keystrokes to enter.

The need for leap seconds arises from fundamentally different considerations. The length of the UTC second is based on the defined frequency of a hyperfine transition in the ground state of the cesium atom (9 192 631 770 Hz). A day constructed using 86,400 of these seconds is somewhat shorter than an astronomical day based on the rotation rate of the earth, which is called UT1.

Clocks based on cesium frequency standards,therefore,run somewhat faster than a clock based on UT1. The difference is about 1 s/year on the average; it is removed by inserting "leap" seconds into the UTC time scale as necessary – in effect stopping the UTC-based clock so that the UT1 scale can catch up. When they are required, these additional seconds are added at the end of June or December. The difference UT1 - UTC therefore exhibits a sawtooth-like behavior – decreasing at a rate of somewhat less than 0.1 s/month for about a year, followed by a step increase of 1.0 s at the time of a leap second. The absolute value of this difference is kept less than 0.9s.

The rate difference UT1 – UTC has been negative since the current leap-second system was introduced in 1972. The magnitude of the rate difference will increase as the earth slows down, but this increase will be somewhat irregular, and it will not be possible to predict exactly when future leap seconds will be needed until a year or so before they are announced. Although "negative" leap seconds are also defined, the fact that the earth will probably continue to slow down (albeit irregularly) suggests that the rate difference between UT1 and UTC will not change sign,so that they will never be needed.

## POSSIBLE SOLUTIONS TO THE ROLLOVER PROBLEM

One way to address the rollover problem is to define a sliding "window" in the software that processes the time-tag. Two-digit years less than or equal to 90, for example, could be assumed to refer to dates in the 21st century, while values of 91 or greater would be assumed to be in the 20th century. This window might be advanced with each revision of the software or whenever another rollover was imminent. The primary advantage of this method is that archived data files need not be changed, nor is any change required to the methods used to enter new records. We are, in effect, carrying part of the date in the time-tag itself and part in the software that interprets it. This method obviously fails for time intervals that span more than one rollover period or for time tags associated with events that are in the distant past or are far in the future. Since rollovers are relatively infrequent in all of the systems we are considering, we would hope that the number of time tags that are affected by these ambiguities would be pretty small. Events that would be affected in this way would be so precious that ancillary means will have to be provided for identifying the rollover value associated with them. If this is not the case, then one of the more general solutions discussed below will be needed.

Although data archives need not be changed, the sliding window solution *does* require a change to the software that processes the time tags, and identifying all of the programs that must be changed can be an enormously expensive and difficult job. If these costs are large enough, it may be worthwhile considering changing the formats of the time-tags themselves at the same time so as to realize a more permanent solution to the problem. One possibility is to change to a 4-digit year, but other solutions are possible that are almost as good and do not require more than the 6 digits currently used by the YYMMDD format. Changing to 3 digits for the year and 3 for the day of the year is one possibility; another would be to use a pure 6-digit day number – perhaps an expanded version of the Modified Julian Day number used by timing laboratories; still another would be to express the date in a base larger than base 10 (although this might break programs that expected only digits in the time-tag). A "shell" program would be needed in each of these cases to translate between the internal format and a more conventional one that was suitable for data entry and display.

The sliding window solution is not perfect, but, at least in the short run, it may be the cheapest one to implement, since most data records will not require modification. The rollover problem will return again near the end of the window period, of course, but the cost of dealing with a relatively infrequent rollover might be more than offset by the savings in file size or program complexity that can be realized in the interim. These savings were much more important in the past when computers had much less memory and much smaller disks. Nevertheless, pushing the cost of converting a large database into the future (when a more elegant solution might be available and when somebody else will pay the price for the conversion) is likely to remain an attractive possibility.

## DEALING WITH LEAP SECONDS

There are two methods that are usually used to account for leap seconds when specifying a time-tag. In the first method, the clock always advances at a constant rate, and leap seconds are incorporated by a separate procedure that "knows" when each leap second has occurred (by means of a separate table, for example) and adjusts the time-tag as necessary. This procedure must intercept all requests for time-tags or time interval calculations and must do the "right thing" both for times in the past and those in the future. Alternatively, the clock itself is corrected for the leap second by applying a time or frequency step each time one occurs. The first method, which is also used to correct for daylight saving time in many systems, simplifies the clock hardware and software, while the second method is simpler to support since it does not require maintaining, distributing and verifying an ancillary table of the dates on which leap seconds have occurred. The second method has difficulties and ambiguities of its own, however, especially in a distributed environment, where different systems may realize the leap second time step in different ways and at slightly different times because they are not perfectly synchronized. Neither method is adequate for computing precise UTC-based time intervals very far in the future, since the times of future leap seconds cannot be accurately predicted. Time-tags of events in the past (before the current leap second system began) have additional ambiguities. These are especially confusing for precisely dating events between 1 January 1958 and 1 January 1972 because UTC was steered *both* in rate and in time during this period.

Leap seconds also introduce ambiguities into time intervals, and systems where time interval (or frequency) is the primary observable usually do not use a UTC-based scale for this reason. The most common contemporary example is GPS time, which keeps track of the number of leap seconds in a separate parameter that is not part of the clock reading itself. (GPS system time itself is not affected by a leap second – only the separate leap-second counters are changed when one occurs. Thus, the difference between GPS system time and UTC changes by 1s each time a leap second occurs. It is up to the receiver to deal with these flags and to include the correction in the output time message when a UTC-based time tag is requested.)

Leap seconds are added after 23:59:59 on the last day of June or December; the extra second is named 23:59:60, and it is followed by 00:00:00 of the next day. This nomenclature may be a problem in some situations. In the first place, a time tag with a value of 60 in the seconds field may be rejected as a format error by some applications. Furthermore, there is no natural way of incorporating leap seconds in formats that do not use hours, minutes and seconds to report the time. The Network Time Protocol falls into this category, as do many other formats that report the time as a number of seconds since some origin time.

There are two alternatives to reporting a time of 23:59:60 during a leap second: the first is to freeze the time at the format-specific equivalent of 23:59:59 for an extra second, and, optionally, to provide a separate flag to indicate that a leap second is in progress during the second of these seconds. This flag is extra baggage, of course, but it cannot be dropped because it is the only way of distinguishing between two physically distinct seconds that have the same time-tag. The second is to slow the frequency of the clock oscillator by a factor of 2 so that it takes two physical seconds to advance from 23:59:59 to 00:00:00. These two systems cannot be used together, of course, since they have very different notions of time-tags while the leap second is in progress: the leap second is identified as 23:59:59 with the ancillary flag set in the first realization and as 23:59:59.5 in the second version. Neither of them is consistent with the "standard" definition of 23:59:60 for the name of the leap second.

Even when the time is transmitted as HH:MM:SS, there are no standards either for how to provide advance notice that a leap second has been scheduled or for how to indicate that one is currently in progress. This notice is important for two reasons. In the first place, it can be used to signal the receiving software that a value of 60 for the seconds is a valid leap second and not a format error. In the second place it can be used to schedule a local procedure to incorporate the leap sėcond into the local time scale in an orderly fashion in whatever way is deemed appropriate (i.e., using a frequency step, a time step or adding an entry to the leap-second table). The greatest ambiguities in how this notice is implemented arise in the IRIG codes, where different manufacturers use different control bits for the leap-second flag. The method used to specify the year in IRIG codes also varies from one implementation to another. Some industries have specified how these parameters shall be defined for the systems that they purchase, but these definitions are not universally recognized.

## VERIFICATION AND TESTING

Whatever solution is adopted, rollovers and leap seconds are relatively rare events, and it is often difficult to verify that a particular system will handle them properly. This is especially serious for rollover events, since neither a century rollover nor a GPS week rollover has happened in the era of computers and digital systems. The performance of GPS receivers when the GPS week number rolls over to 0 can be evaluated using a simulator, and most of the newer receivers handle the rollover properly using some form of the sliding-window algorithm discussed above. Some older receivers may fail these tests, and it may be difficult or impossible to repair them. Some receivers will only fail in the immediate vicinity of the rollover because they will try to use a pre-rollover almanac to estimate the post-rollover positions of the satellites. Although the rollover ambiguity will mean that a conversion between GPS time and UTC always will be wrong by 1024 weeks following the rollover event, many of these older receivers will start tracking normally again after the rollover once a post-rollover almanac has been received. Although repairing software systems is possible in principle, the complexity of the programs and the inadequacy of the documentation may make this difficult (and therefore very expensive) in practice.

Radio and GPS receivers must also be tested to be sure that they recognize the flag giving advance notice of a leap second and that they respond properly during the event. This is not always a simple matter, since there may be a complicated interaction between the leap second logic and the "holdover" algorithm that controls the internal oscillator if the external time signal is temporarily lost. (A similar ambiguity may arise when different GPS satellites are transmitting different values for the leap second flags.) In a worst-case approach, it may be prudent to ignore the time transmitted by a radio- or GPS-synchronized clock for some time after the leap second has occurred to give the receiver time to re-synchronize itself.

## CONCLUSIONS

Rollover problems in digital time formats arise from engineering compromises between size, speed, time and money. While rollover events cannot be eliminated in a digital format, they can be pushed arbitrarily far into the future by making the time code longer or by using a shorter time code with ancillary flags to provide rollover information. This compromise cannot be the sole province of the system designer – the users must also play a role in formulating the balance between the savings that can be realized using a smaller time-tag and the costs of dealing with the more frequent rollovers that will result from this choice.

The problem of leap seconds is more complicated and the solutions are, therefore, more ambiguous. There is no simple way of simultaneously accommodating users who want an atomic-based time scale that is smooth and continuous and those who want one that closely approximates a time-scale based on astronomical observations. No single system can meet all of these requirements, and multiple systems will coexist for the foreseeable future as a result. Understanding how to convert among these systems will continue to be a guarantee of employment for some and a source of confusion for many others.

Finally, it would be useful to adopt standard methods for all aspects of a time code, including the relationship between the time-tag and the on-time marker and the method used to provide information about future and pending leap seconds.