Date & timestamp guidelines

By Air Sensor Workgroup

Status of this document

The Air Sensor Workgroup (ASW) adopted the date and timestamp guidelines on 17 January 2017.

Abstract

This document defines the Date and Timestamp Guidelines for use in the field of air quality measurement and monitoring. It has been derived from ISO 8601 standard, IETF RFC 3339 and the W3C profile.

Users and applicability

ASW strongly encourages the manufacturers of low cost sensors, researchers working on air quality and any other air quality data generators and users to use these guidelines.

Purpose of guidelines

Numerous individuals and organizations across the globe have spent effort to measure air quality data and determine the impact of air pollution on human health. However, most of them have been isolated efforts. ASW sees tremendous value in sharing data across data owners so that researchers and other interested parties can take advantage of the vast amount of data to create air quality data products that can help communities worldwide. To this effect, ASW has been developing standards for data generation, storage and exchange.

The Date and Timestamp guidelines apply to data generation and storage by the sensors and their backend database systems, thus facilitating accurate, reliable and efficient exchange of data across various data owners and data users.

Format for data generation

Use Epoch time (aka Unix time, POSIX time) which is time in seconds since Unix Epoch (1970-01-01T00:00:00Z) as a 64-bit unsigned integer at the point of generating date/time value by the device.

Notes:

1. Epoch time is UTC based and hence, does not have the complexity of time zones; however it can easily be converted to any local timezone when necessary
2. It is an integer and hence needs less storage, particularly helpful as the sensors have limited compute capacity. It also minimizes the size of data transmitted back to the server (via SMS, Wi-Fi, etc.).
3. It simplifies date arithmetic
4. Use 64-bit integer as the data type so as to avoid data overflow in the year 2038
5. Use microseconds granularity in order to support measurements at frequencies higher than once per second (1Hz)
6. Use this format for transmitting data from the sensor to the backend server
7. For writing to the logs, including logs stored on the sensor or the sensor system, use the human readable format and associated guidelines described in the next section. This increases the ease and efficiency of onsite troubleshooting and maintenance.
8. Tools and libraries to convert to human-readable formats are available and can be applied just prior to visualization

Key considerations

- Multiple application tiers on the device such as I2C/IC/RTC, microprocessor and software may generate the timestamp value. All these tiers must support 64-bit integer. If any of these support 32-bit only, then the Epoch time will overflow on 19-Jan-2038 and reset to 13-Dec-1901.
- Some RTCs have a ceiling on the year they support like 2099, 2100, etc. Pay attention to what is supported and have a plan for subsequent time period.
- Use appropriate data type in the software programs to support 64-bit integer values
- If there is a need for sub-second measurements, explore the built-in support provided by the databases and programming languages (see Appendix).

Format for data storage

The requirements for data storage are driven not only by storage optimization and retrieval (I/O) performance considerations but also the subsequent data usage, including the visualization layer. Store the timestamp value as Epoch time (as received from the sensor) as well as in a human readable format. The Epoch time can be used by the downstream applications such as modeling or other data processing systems while the human readable format can be used directly by the visualization layer such as reports and dashboards. By storing in both formats, the processing burden on the visualization layer can be reduced or avoided.

Human readable timestamp

For the human readable format, convert the Epoch time received from the sensor to the local time adjusted for timezone and Daylight Saving Time (if applicable) based on the location of the sensor and store using the ISO 8601 format as shown below. Use the timezone offset so that this data value is self-contained.

Format: YYYY-MM-DDThh:mm:ss.nnn±hh:mm
Example: 2016-08-25T15:23:22.635-07:00

Notes

1. The above format is adapted from ISO 8601 standard, IETF RFC 3339 and the W3C profile.
2. The ISO 8601 standard offers an option to omit the usage of [T] as time designator between the date & time components of the timestamp. However, we strongly recommend using [T] to proactively address potential data processing issues that may arise due to the presence of a whitespace character; this is especially important in the Hadoop/Big Data ecosystem.
3. Represent midnight as 00:00:00 (and not as 24:00:00) to simplify data processing
4. Store the human readable format of timestamp in the local timezone adjusted for DST where the measurement was recorded in order to avoid storing redundant data
5. The Epoch time represents UTC and the human readable format represents the local timezone, thus covering both variations in the data storage layer

Appendix

1. A summary of the international standard date and time notation -- https://www.cl.cam.ac.uk/~mgk25/iso-time.html

   Some hints on using ISO 8601 in software

   - The ISO 8601 notation is today the commonly recommended format of representing date and time as human-readable strings in new plain-text communication protocols and file formats. Several standards and profiles have been derived from ISO 8601, including RFC 3339 and a W3C note on date and time formats.

   - The C and POSIX standards define for the strftime() function and the date utility a notation for defining date and time representations. Here are some examples, of how they can be used to produce ISO 8601 output:

Format string	Output
%Y-%m-%d	1999-12-31
%Y-%j	1999-365
%G-W%V-%u	1999-W52-5
%H:%M:%S	23:59:59

   Other links about date, time, and calendars

   - Some other interesting sources of information about date and time on the Internet are for example the Glossary of Frequency and Timing Terms and the FAQ provided by NIST, the Yahoo Science:Measurements and Units:Time link collection, the U.S. Naval Observatory Server, the International Earth Rotation Service (IERS), the Network Time Protocol (NTP), the time and calendar section of the USENET sci.astro FAQ, and the Calendar FAQ.

2. History:

    - IETF RFC7231 - https://yada.juxt.pro/spec/rfc7231#page-65

3. POSIX/Epoch reference:

    - http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7542098
    - http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6581844&queryText=posix%20time&refinements=4294965216
    - http://lost.co.nz/programming/epoch.html

4. Methods to get sub-second timestamp since Epoch:

    Unix: `date +%s%3N` ⟶ provides milliseconds since Unix epoch

    http://unix.stackexchange.com/questions/69322/how-to-get-milliseconds-since-unix-epoch
    Unix: `date --rfc-3339=ns` ⟶ provides timestamp in nanoseconds in ISO 8601 format

    `date --rfc-3339=ns | sed 's/ /T/; s/\(\....\).*-/\1-/g'` ⟶ adds the T designator between date and time

    `date --iso-8601=ns` ⟶ performs slower than --rfc-3339 option
    http://unix.stackexchange.com/questions/120484/what-is-a-standard-command-for-printing-a-date-in-rfc-3339-format

5. Reference for various databases and programming languages: https://currentmillis.com/

6. Benefits of Data Standards, EPA - https://www.epa.gov/data-standards/learn-about-data-standards