Sources for time zone and daylight saving time data

Time zone and daylight-saving rules are controlled by individual governments. They are sometimes changed with little notice, and their histories and planned futures are often recorded only fitfully. Here is a summary of attempts to organize and record relevant data in this area.

Outline

The tz database

The public-domain time zone database contains code and data that represent the history of local time for many representative locations around the globe. It is updated periodically to reflect changes made by political bodies to time zone boundaries and daylight saving rules. This database (known as `tz`, `tzdb`, or `zoneinfo`) is used by several implementations, including the GNU C Library (used in GNU/Linux), Android, FreeBSD, NetBSD, OpenBSD, Chromium OS, Cygwin, MariaDB, MINIX, MySQL, webOS, AIX, BlackBerry 10, iOS, macOS, Microsoft Windows, OpenVMS, Oracle Database, and Oracle Solaris.

Each main entry in the database represents a *timezone* for a set of civil-time clocks that have all agreed since 1970. Timezones are typically identified by continent or ocean and then by the name of the largest city within the region containing the clocks. For example, `America/New_York` represents most of the US eastern time zone; `America/Phoenix` represents most of Arizona, which uses mountain time without daylight saving time (DST); `America/Detroit` represents most of Michigan, which uses eastern time but with different DST rules in 1975; and other entries represent smaller regions like Starke County, Indiana, which switched from central to eastern time in 1991 and switched back in 2006. To use the database on an extended POSIX implementation set the `TZ` environment variable to the location's full name, e.g., `TZ="America/New_York"`.

Associated with each timezone is a history of offsets from Universal Time (UT), which is Greenwich Mean Time (GMT) with days beginning at midnight; for timestamps after 1960 this is more precisely Coordinated

Universal Time (UTC). The database also records when daylight saving time was in use, along with some time zone abbreviations such as EST for Eastern Standard Time in the US.

Downloading the tz database

The following shell commands download the latest release's two tarballs to a GNU/Linux or similar host.

```
mkdir tzdb
cd tzdb
wget https://www.iana.org/time-zones/repository/tzcode-latest.tar.gz
wget https://www.iana.org/time-zones/repository/tzdata-latest.tar.gz
gzip -dc tzcode-latest.tar.gz | tar -xf -
gzip -dc tzdata-latest.tar.gz | tar -xf -
```

Alternatively, the following shell commands download the same release in a single-tarball format containing extra data useful for regression testing:

```
wget https://www.iana.org/time-zones/repository/tzdb-latest.tar.lz
lzip -dc tzdb-latest.tar.lz | tar -xf -
```

These commands use convenience links to the latest release of the tz database hosted by the Time Zone Database website of the Internet Assigned Numbers Authority (IANA). Older releases are in files named tzcode*V*.tar.gz, tzdata*V*.tar.gz, and tzdb-*V*.tar.lz, where *V* is the version. Since 1996, each version has been a four-digit year followed by lower-case letter (a through z, then za through zz, then zza through zzz, and so on). Since version 2016h, each release has contained a text file named "version" whose first (and currently only) line is the version. Older releases are archived, and are also available in an FTP directory via a less-secure protocol.

Alternatively, a development repository of code and data can be retrieved from GitHub via the shell command:

```
git clone https://github.com/eggert/tz
```

Since version 2012e, each release has been tagged in development repositories. Untagged commits are less well tested and probably contain more errors.

After obtaining the code and data files, see the README file for what to do next. The code lets you compile the tz source files into machine-readable binary files, one for each location. The binary files are in a special timezone information format (*TZif*) specified by Internet RFC 8536. The code also lets you read a TZif file and interpret timestamps for that location.

Changes to the tz database

The tz code and data are by no means authoritative. If you find errors, please send changes to tz@iana.org, the time zone mailing list. You can also subscribe to it and browse the archive of old messages. Metadata for mailing list discussions and corresponding data changes can be generated automatically.

If your government plans to change its time zone boundaries or daylight saving rules, inform tz@iana.org well in advance, as this will coordinate updates to many cell phones, computers, and other devices around the world. With less than a year's notice there is a good chance that some computer-based clocks will operate incorrectly after the change, due to delays in propagating updates to software and data. The shorter the notice, the more likely clock problems will arise; see "On the Timing of Time Zone Changes" for examples.

Changes to the tz code and data are often propagated to clients via operating system updates, so client tz data can often be corrected by applying these updates. With GNU/Linux and similar systems, if your maintenance provider has not yet adopted the latest tz data, you can often short-circuit the process by tailoring the generic instructions in the tz README file and installing the latest data yourself. System-specific instructions for installing

the latest tz data have also been published for AIX, Android, ICU, IBM and Oracle Java, Joda-Time, MySQL, and Noda Time (see below).

Sources for the tz database are UTF-8 text files with lines terminated by LF, which can be modified by common text editors such as GNU Emacs, gedit, and vim. Specialized source-file editing can be done via the Sublime zoneinfo package for Sublime Text and the VSCode zoneinfo extension for Visual Studio Code.

For further information about updates, please see Procedures for Maintaining the Time Zone Database (Internet RFC 6557). More detail can be found in Theory and pragmatics of the tz code and data. A0 TimeZone Migration displays changes between recent tzdb versions.

Commentary on the tz database

- The article tz database is an encyclopedic summary.
- How to Read the tz Database Source Files explains the tz database format.
- A literary appreciation of the Olson/Zoneinfo/tz database comments on the database's style.

Web sites using recent versions of the tz database

These are listed roughly in ascending order of complexity and fanciness.

- Time.is shows locations' time and zones.
- TimeJones.com, Time Zone Converter and The World Clock are time zone converters.
- Date and Time Gateway lets you see the TZ values directly.
- Current Time in 1000 Places uses descriptions of the values.
- Time Zone Converter uses a pulldown menu.
- Complete timezone information for all countries displays tables of DST rules.
- The World Clock – Worldwide lets you sort zone names and convert times.
- 24TimeZones has a world time map and a time converter.
- Time Difference calculates the current time difference between locations.
- Weather Now and The Time Now list the weather too.

Network protocols for tz data

- The Internet Engineering Task Force's Time Zone Data Distribution Service (tzdist) working group defined TZDIST (Internet RFC 7808), a time zone data distribution service, along with CalDAV (Internet RFC 7809), a calendar access protocol for transferring time zone data by reference. The tzdist-bis mailing list discussed the Internet draft TZDIST Geolocate Extension (now expired), which let a client determine its timezone from its geographic location using a 'geo' URI.
- The Internet Calendaring and Scheduling Core Object Specification (iCalendar) (Internet RFC 5445) covers time zone data; see its VTIMEZONE calendar component. The iCalendar format requires specialized parsers and generators; a variant xCal (Internet RFC 6321) uses XML format, and a variant jCal (Internet RFC 7265) uses JSON format.

Other tz compilers

Although some of these do not fully support tz data, in recent tzdb distributions you can generally work around compatibility problems by running the command `make rearguard_tarballs` and compiling from the resulting tarballs instead.

- Vzic is a C program that compiles tz source into iCalendar-compatible VTIMEZONE files. Vzic is freely available under the GNU General Public License (GPL).

- tziCal – tz database conversion utility is like Vzic, except for the .NET framework and with a BSD-style license.
- DateTime::TimeZone contains a script `parse_olson` that compiles `tz` source into Perl modules. It is part of the Perl DateTime Project, which is freely available under both the GPL and the Perl Artistic License. DateTime::TimeZone also contains a script `tests_from_zdump` that generates test cases for each clock transition in the `tz` database.
- The Time Zone Database Parser is a C++ parser and runtime library with API adopted into the draft standard for C++20, the next iteration of the C++ standard. It is freely available under the MIT license.
- International Components for Unicode (ICU) contains C/C++ and Java libraries for internationalization that has a compiler from `tz` source and from CLDR data (mentioned below) into an ICU-specific format. ICU is freely available under a BSD-style license.
- The Tzdata package for the Elixir language downloads and compiles `tz` source and exposes APIs for use. It is freely available under the MIT license.
- Java-based compilers and libraries include:
  - The TZUpdater tool compiles `tz` source into the format used by Oracle Java.
  - The Java SE 8 Date and Time API can be supplemented by ThreeTen-Extra, which is freely available under a BSD-style license.
  - Joda-Time – Java date and time API contains a class `org.joda.time.tz.ZoneInfoCompiler` that compiles `tz` source into a binary format. It inspired Java 8 `java.time`, which its users should migrate to once they can assume Java 8 or later. It is available under the Apache License.
  - IANA Updater, tzdbgen, and ZIUpdater are other alternatives to TZUpdater. IANA Updater's license is unclear; the others are licensed under the GPL.
  - Time4A: Advanced date and time library for Android and Time4J: Advanced date, time and interval library for Java compile `tz` source into a binary format. Time4A is available under the Apache License and Time4J is available under the GNU Lesser General Public License (LGPL).
  - ICU (mentioned above) contains compilers and Java-based libraries.
- Noda Time – Date and time API for .NET is like Joda-Time and Time4J, but for the .NET framework instead of Java. It is freely available under the Apache License.
- JavaScript-based compilers and libraries include:
  - CompactTimeZoneGenerator compiles time zone data into a compact form designed for JavaScript. It is freely available under a combination of the MIT license and the Apache License.
  - Moment Timezone is a plugin for the Moment.js date manipulation library. It is freely available under the MIT license.
  - TimezoneJS.Date's API is upward compatible with standard JavaScript Dates. It is freely available under the Apache License.
- JuliaTime contains a compiler from `tz` source into Julia. It is freely available under the MIT license.
- TZDB – IANA Time Zone Database for Delphi/FPC compiles from `tz` source into Object Pascal as compiled by Delphi and FPC. It is freely available under a BSD-style license.
- pytz – World Timezone Definitions for Python compiles `tz` source into Python. It is freely available under a BSD-style license.
- TZInfo – Ruby Timezone Library compiles `tz` source into Ruby. It is freely available under the MIT license.
- The Chronos Date/Time Library is a Smalltalk class library that compiles `tz` source into a time zone repository whose format is either proprietary or an XML-encoded representation.
- Tcl contains a developer-oriented parser that compiles `tz` source into text files, along with a runtime that can read those files. Tcl is freely available under a BSD-style license.

Other TZif readers

- The GNU C Library has an independent, thread-safe implementation of a TZif file reader. This library is freely available under the LGPL and is widely used in GNU/Linux systems.
- GNOME's GLib has a TZif file reader written in C that creates a `GTimeZone` object representing sets of UT offsets. It is freely available under the LGPL.

- The BDE Standard Library's `baltzo::TimeZoneUtil` component contains a C++ implementation of a TZif file reader. It is freely available under the Apache License.
- CCTZ is a simple C++ library that translates between UT and civil time and can read TZif files. It is freely available under the Apache License.
- ZoneInfo.java is a TZif file reader written in Java. It is freely available under the LGPL.
- Timelib is a C library that reads TZif files and converts timestamps from one time zone or format to another. It is used by PHP, HHVM, and MongoDB. It is freely available under the MIT license.
- Timezone is a JavaScript library that supports date arithmetic that is time zone aware. It is freely available under the MIT license.
- Tcl, mentioned above, also contains a TZif file reader.
- DateTime::TimeZone::Tzfile is a TZif file reader written in Perl. It is freely available under the same terms as Perl (dual GPL and Artistic license).
- The public-domain tz.js library contains a Python tool that converts TZif data into JSON-format data suitable for use in its JavaScript library for time zone conversion. Dates before 1970 are not supported.
- The timezone-olson package contains Haskell code that parses and uses TZif data. It is freely available under a BSD-style license.

Other `tz`-based time zone software

- FoxClocks is an extension for Google Chrome and for Mozilla Toolkit applications like Firefox and Thunderbird. It displays multiple clocks in the application window, and has a mapping interface to Google Earth. It is freely available under the GPL.
- Go programming language implementations contain a copy of a 32-bit subset of a recent `tz` database in a Go-specific format.
- International clock (intclock) is a clock that displays multiple time zones on GNU/Linux and similar systems. It is freely available under the GPL.
- Microsoft Windows 8.1 and later has `tz` data and CLDR data (mentioned below) used by the Windows Runtime / Universal Windows Platform classes `DateTimeFormatter` and `Calendar`. Exploring Windows Time Zones with `System.TimeZoneInfo` describes the older, proprietary method of Microsoft Windows 2000 and later, which stores time zone data in the Windows Registry. The Zone → Tzid table or XML file of the CLDR data maps proprietary zone IDs to `tz` names. These mappings can be performed programmatically via the TimeZoneConverter .NET library, or the ICU Java and C++ libraries mentioned above.
- Oracle Java contains a copy of a subset of a recent `tz` database in a Java-specific format.
- Time Zone Master is a Microsoft Windows clock program that can automatically download, compile and use `tz` releases. The Basic version is free.
- VelaTerra is a macOS program. Its developers offer free licenses to `tz` contributors.

Other time zone databases

- Time-zone Atlas is Astrodienst's Web version of Shanks and Pottenger's out-of-print time zone history atlases for the US and for the world, now published in software form by ACS-Starcrafts. Although these extensive atlases were sources for much of the older `tz` data, they are unreliable as Shanks appears to have guessed many UT offsets and transitions. The atlases cite no sources and do not indicate which entries are guesswork.
- HP-UX has a database in its own `tztab`(4) format.
- Microsoft Windows has proprietary data mentioned above.
- World Time Server is another time zone database.
- The Standard Schedules Information Manual of the International Air Transport Association gives current time zone rules for airports served by commercial aviation.

Maps

- The United States Central Intelligence Agency (CIA) publishes a time zone map; the Perry–Castañeda Library Map Collection of the University of Texas at Austin has copies of recent editions. The pictorial quality is good, but the maps do not indicate daylight saving time, and parts of the data are a few years out of date.
- World Time Zone Map with current time has several fancy time zone maps; it covers Russia particularly well. The maps' pictorial quality is not quite as good as the CIA's but the maps are more up to date.
- How much is time wrong around the world? maps the difference between mean solar and standard time, highlighting areas such as western China where the two differ greatly. It's a bit out of date, unfortunately.

Time zone boundaries

Geographical boundaries between timezones are available from several geolocation services and other sources.

- Timezone Boundary Builder extracts Open Street Map data to build boundaries of `tzdb` timezones. Its code is freely available under the MIT license, and its data entries are freely available under the Open Data Commons Open Database License. The maps' borders appear to be quite accurate.
- Programmatic interfaces that map geographical coordinates via tz_world to `tzdb` timezones include:
  - GeoTimeZone is written in C# and is freely available under the MIT license.
  - The latlong package is written in Go and is freely available under the Apache License.
  - LatLongToTimezone, in both Java and Swift form, is freely available under the MIT license.
  - For Node.js, the geo-tz module is freely available under the MIT license, and the tz-lookup module is in the public domain.
  - The timezonefinder library for Python is freely available under the MIT license.
  - The timezone_finder library for Ruby is freely available under the MIT license.
- Free access via a network API, if you register a key, is provided by the GeoNames Timezone web service, the Google Maps Time Zone API, and the Time Zone Database & API. Commercial network API access is provided by AskGeo and GeoGarage.
- "How to get a time zone from a location using latitude and longitude coordinates?" discusses other geolocation possibilities.
- Administrative Divisions of Countries ("Statoids") lists political subdivision data related to time zones.
- Time zone boundaries for multizone countries summarizes legal boundaries between time zones within countries.
- Manifold Software – GIS and Database Tools includes a Manifold-format map of world time zone boundaries distributed under the GPL.
- A ship within the territorial waters of any nation uses that nation's time. In international waters, time zone boundaries are meridians 15° apart, except that UT−12 and UT+12 are each 7.5° wide and are separated by the 180° meridian (not by the International Date Line, which is for land and territorial waters only). A captain can change ship's clocks any time after entering a new time zone; midnight changes are common.

Civil time concepts and history

- A Walk through Time surveys the evolution of timekeeping.
- The history of daylight saving time is surveyed in About Daylight Saving Time – History, rationale, laws & dates and summarized in A Brief History of Daylight Saving Time.
- Time Lords discusses how authoritarians manipulate civil time.
- Working with Time Zones contains guidelines and best practices for software applications that deal with civil time.
- A History of the International Date Line tells the story of the most important time zone boundary.
- Basic Time Zone Concepts discusses terminological issues behind time zones.

National histories of legal time

Australia
: The Parliamentary Library commissioned a research paper on daylight saving time in Australia. The Bureau of Meteorology publishes a list of Implementation Dates of Daylight Savings Time within Australia.

Belgium
: The Royal Observatory of Belgium maintains a table of time in Belgium (in Dutch and French).

Brazil
: The Time Service Department of the National Observatory records Brazil's daylight saving time decrees (in Portuguese).

Canada
: National Research Council Canada publishes current and some older information about time zones and daylight saving time.

Chile
: The Hydrographic and Oceanographic Service of the Chilean Navy publishes a history of Chile's official time (in Spanish).

China
: The Hong Kong Observatory maintains a history of summer time in Hong Kong, and Macau's Meteorological and Geophysical Bureau maintains a similar history for Macau. Unfortunately the latter is incomplete and has errors.

Czech Republic
: When daylight saving time starts and ends (in Czech) summarizes and cites historical DST regulations.

Germany
: The National Institute for Science and Technology maintains the Realisation of Legal Time in Germany.

Israel
: The Interior Ministry periodically issues announcements (in Hebrew).

Italy
: The National Institute of Metrological Research publishes a table of civil time (in Italian).

Malaysia
: See Singapore below.

Mexico
: The Investigation and Analysis Service of the Mexican Library of Congress has published a history of Mexican local time (in Spanish).

Netherlands
: Legal time in the Netherlands (in Dutch) covers the history of local time in the Netherlands from ancient times.

New Zealand
: The Department of Internal Affairs maintains a brief History of Daylight Saving.

Singapore
: Why is Singapore in the "Wrong" Time Zone? details the history of legal time in Singapore and Malaysia.

United Kingdom
: History of legal time in Britain discusses in detail the country with perhaps the best-documented history of clock adjustments.

United States
: The Department of Transportation's Recent Time Zone Proceedings lists changes to time zone boundaries.

Uruguay
: The Oceanography, Hydrography, and Meteorology Service of the Uruguayan Navy (SOHMA) publishes an annual almanac (in Spanish).

Costs and benefits of time shifts

Various sources argue for and against daylight saving time and time zone shifts, and many scientific studies have been conducted. This section summarizes reviews of scientific literature in the area.

- Carey RN, Sarma KM. Impact of daylight saving time on road traffic collision risk: a systematic review. *BMJ Open.* 2017;7(6):e014319. doi:10.1136/bmjopen-2016-014319. This reviews research literature and concludes that the evidence neither supports nor refutes road safety benefits from shifts in time zones.
- Havranek T, Herman D, Irsova D. Does daylight saving save electricity? A meta-analysis. *Energy J.* 2018;39(2). doi:10.5547/01956574.39.2.thav. This analyzes research literature and concludes, "Electricity savings are larger for countries farther away from the equator, while subtropical regions consume more electricity because of DST."
- Roenneberg T, Winnebeck EC, Klerman EB. Daylight saving time and artificial time zones – a battle between biological and social times. *Front Physiol.* 2019;10:944. doi:10.3389/fphys.2019.00944. This reviews evidence about the health effects of DST and concludes, "In summary, the scientific literature strongly argues against the switching between DST and Standard Time and even more so against adopting DST permanently."

Precision timekeeping

- The Science of Timekeeping is a thorough introduction to the theory and practice of precision timekeeping.
- The Science of Time 2016 contains several freely-readable papers.
- NTP: The Network Time Protocol (Internet RFC 5905) discusses how to synchronize clocks of Internet hosts.
- The Huygens family of software algorithms can achieve accuracy to a few tens of nanoseconds in scalable server farms without special hardware.
- The Precision Time Protocol (IEEE 1588) can achieve submicrosecond clock accuracy on a local area network with special-purpose hardware.
- Timezone Options for DHCP (Internet RFC 4833) specifies a DHCP option for a server to configure a client's time zone and daylight saving settings automatically.
- Astronomical Times explains more abstruse astronomical time scales like TDT, TCG, and TDB. Time Scales goes into more detail, particularly for historical variants.
- The IAU's SOFA collection contains C and Fortran code for converting among time scales like TAI, TDB, TDT and UTC.
- Mars24 Sunclock – Time on Mars describes Airy Mean Time (AMT) and the diverse local time scales used by each landed mission on Mars.
- LeapSecond.com is dedicated not only to leap seconds but to precise time and frequency in general. It covers the state of the art in amateur timekeeping, and how the art has progressed over the past few decades.
- The rules for leap seconds are specified in Annex 1 (Time scales) of Standard-frequency and time-signal emissions, International Telecommunication Union – Radiocommunication Sector (ITU-R) Recommendation TF.460-6 (02/2002).
- IERS Bulletins contains official publications of the International Earth Rotation and Reference Systems Service, which decides when leap seconds occur. The `tz` code and data support leap seconds via an optional "`right`" configuration, as opposed to the default "`posix`" configuration.
- Leap Smear discusses how to gradually adjust POSIX clocks near a leap second so that they disagree with UTC by at most a half second, even though every POSIX minute has exactly sixty seconds. This approach works with the default `tz` "`posix`" configuration, is supported by the NTP reference implementation, and is used by major cloud service providers. However, according to §3.7.1 of Network Time Protocol Best Current Practices (Internet RFC 8633), leap smearing is not suitable for applications requiring accurate UTC or civil time, and is intended for use only in single, well-controlled environments.
- The Leap Second Discussion List covers McCarthy and Klepczynski's 1999 proposal to discontinue leap seconds, discussed further in The leap second: its history and possible future. UTC might be redefined without Leap Seconds gives pointers on this contentious issue, which was active until 2015 and could become active again.

Time notation

- The Unicode Common Locale Data Repository (CLDR) Project has localizations for time zone names, abbreviations, identifiers, and formats. For example, it contains French translations for "Eastern European Summer Time", "EEST", and "Bucharest". Its by-type charts show these values for many locales. Data values are available in both LDML (an XML format) and JSON.
- A summary of the international standard date and time notation covers *ISO 8601-1:2019 – Date and time – Representations for information interchange – Part 1: Basic rules*.
- XML Schema: Datatypes – dateTime specifies a format inspired by ISO 8601 that is in common use in XML data.
- §3.3 of Internet Message Format (Internet RFC 5322) specifies the time notation used in email and HTTP headers.
- Date and Time on the Internet: Timestamps (Internet RFC 3339) specifies an ISO 8601 profile for use in new Internet protocols.
- Date & Time Formats on the Web surveys web- and Internet-oriented date and time formats.
- Alphabetic time zone abbreviations should not be used as unique identifiers for UT offsets as they are ambiguous in practice. For example, in English-speaking North America "CST" denotes 6 hours behind UT, but in China it denotes 8 hours ahead of UT, and French-speaking North Americans prefer "HNC" to "CST". The `tz` database contains English abbreviations for many timestamps; unfortunately some of these abbreviations were merely the database maintainers' inventions, and these have been removed when possible.
- Numeric time zone abbreviations typically count hours east of UT, e.g., +09 for Japan and −10 for Hawaii. However, the POSIX `TZ` environment variable uses the opposite convention. For example, one might use `TZ="JST-9"` and `TZ="HST10"` for Japan and Hawaii, respectively. If the `tz` database is available, it is usually better to use settings like `TZ="Asia/Tokyo"` and `TZ="Pacific/Honolulu"` instead, as this should avoid confusion, handle old timestamps better, and insulate you better from any future changes to the rules. One should never set POSIX `TZ` to a value like `"GMT-9"`, though, since this would incorrectly imply that local time is nine hours ahead of UT and the time zone is called "GMT".

See also

- Theory and pragmatics of the `tz` code and data
- Time and the Arts

Please send corrections to this web page to the time zone mailing list.