Stream: Internet Engineering Task Force (IETF)
RFC: 9636
Obsoletes: 8536
Category: Standards Track
Published: October 2024
ISSN: 2070-1721
Authors: A. Olson | P. Eggert, *UCLA* | K. Murchison, *Fastmail*

RFC 9636
The Time Zone Information Format (TZif)

Abstract

This document specifies the Time Zone Information Format (TZif) for representing and exchanging time zone information, independent of any particular service or protocol. Two media types for this format are also defined.

This document replaces and obsoletes RFC 8536.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9636.

Copyright Notice

Table of Contents

1. Introduction 3
2. Conventions Used in This Document 3
3. The Time Zone Information Format (TZif) 5
   3.1. TZif Header 6
   3.2. TZif Data Block 7
   3.3. TZif Footer 11
      3.3.1. All-Year Daylight Saving Time 11
      3.3.2. TZ String Extension 12
4. Interoperability Considerations 12
5. Internationalization Considerations 13
6. Use with the Time Zone Data Distribution Service 14
   6.1. Truncating TZif Files 14
   6.2. Example TZDIST Request for TZif Data 15
7. Security Considerations 16
8. Privacy Considerations 16
9. IANA Considerations 16
   9.1. application/tzif 16
   9.2. application/tzif-leap 17
10. References 18
   10.1. Normative References 18
   10.2. Informative References 19
Appendix A. Common Interoperability Issues 20
Appendix B. Example TZif Files 22
   B.1. Version 1 File Representing UTC (with Leap Seconds) 22
   B.2. Version 2 File Representing Pacific/Honolulu 27
   B.3. Truncated Version 2 File Representing Pacific/Johnston 33
   B.4. Truncated Version 3 File Representing Asia/Jerusalem 37
   B.5. Truncated Version 4 File Representing Europe/London 39

Appendix C. Changes from RFC 8536 42
Acknowledgments 43
Authors' Addresses 43

1. Introduction

Time zone data typically consists of offsets from universal time (UT), daylight saving transition rules, one or more local time designations (acronyms or abbreviations), and optional leap-second adjustments. One such format for conveying this information is iCalendar [RFC5545]. It is a text-based format used by calendaring and scheduling systems.

This document specifies the widely deployed Time Zone Information Format (TZif). It is a binary format used by most UNIX systems to calculate local time. This format was introduced in the 1980s and has evolved since then into multiple upward-compatible versions. There is a wide variety of interoperable software capable of generating and reading files in this format [tz-link].

This specification does not define the source of the data assembled into a TZif file. One such source is the IANA-hosted time zone database [RFC6557].

This document obsoletes [RFC8536], providing editorial improvements, new details, and errata fixes while keeping full compatibility with the interchange format of [RFC8536]. Additionally, a new version of the format is defined. The changes from [RFC8536] are summarized in Appendix C.

2. Conventions Used in This Document

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**NOT RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

The following terms are used in this document (see "Time zone and daylight saving time data" [tz-link] for more detailed information about civil timekeeping data and practice):

Coordinated Universal Time (UTC): The basis for civil time since 1960. It is approximately equal to mean solar time at the prime meridian (0 degrees longitude).

Daylight Saving Time (DST): The time according to a location's law or practice, when adjusted as necessary from standard time. The adjustment may be positive or negative, and the amount of adjustment may vary depending on the date and time; the TZif format even allows the adjustment to be zero, although this is not common practice.

International Atomic Time (TAI): The time standard based on atomic clocks since 1972. It is equal to UTC but without leap-second adjustments.

Leap Second: A one-second adjustment to keep UTC close to mean solar time at the prime meridian (see [ITU-R-TF.460]). Each inserted or deleted leap second occurs at the end of a UTC month, that is, a month using the Gregorian calendar and the UTC timescale.

Leap-Second Correction (LEAPCORR): The value of "TAI - UTC - 10" for timestamps after the first leap second, and zero for timestamps before that. The expression "TAI - UTC - 10" comes from the fact that TAI - UTC was defined to be 10 just prior to the first leap second in 1972, so clocks with leap seconds have a zero LEAPCORR before the first leap second.

Local Time: Civil time for a particular location. Its offset from universal time can depend on the date and time of day.

POSIX Epoch: 1970-01-01 00:00:00 UTC, the basis for absolute timestamps in this document.

Standard Time: The time according to a location's law or practice, unadjusted for daylight saving time.

Time Change: A change to civil timekeeping practice. It occurs when one or more of the following happen simultaneously:

1. a change in UT offset
2. a change in whether daylight saving time is in effect
3. a change in time zone abbreviation
4. a leap second (i.e., a change in LEAPCORR)

Time Zone Data: The Time Zone Data Distribution Service (TZDIST) [RFC7808] defines "Time zone data" as "data that defines a single time zone, including an identifier, UTC offset values, DST rules, and other information such as time zone abbreviations". The interchange format defined in this document is one such form of time zone data.

Transition Time: The moment of occurrence of a time change that is not a leap second. It is identified with a signed integer count of UNIX leap time seconds since the POSIX epoch.

Universal Time (UT): The basis of civil time. This is the principal form of the mean solar time at the prime meridian (0 degrees longitude) for timestamps before UTC was introduced in 1960 and is UTC for timestamps thereafter. Although UT is sometimes called "UTC" or "GMT" in other sources, this specification uses the term "UT" to avoid confusion with UTC or with GMT.

UNIX Time: The time as returned by the time() function provided by the C programming language (see Section 3 of the "System Interfaces" volume of [POSIX]). This is an integer number of seconds since the POSIX epoch, not counting leap seconds. As an extension to POSIX, negative values represent times before the POSIX epoch, using UT.

UNIX Leap Time: UNIX time plus all preceding leap-second corrections. For example, if the first leap-second record in a TZif file occurs at 1972-06-30 23:59:60 UTC, the UNIX leap time for the timestamp 1972-07-01 00:00:00 UTC would be 78796801, one greater than the UNIX time for the same timestamp. Similarly, if the second leap-second record occurs at 1972-12-31 23:59:60

UTC, it accounts for the first leap second, so the UNIX leap time of 1972-12-31 23:59:60 UTC would be 94694401, and the UNIX leap time of 1973-01-01 00:00:00 UTC would be 94694402. If a TZif file specifies no leap-second records, UNIX leap time is equal to UNIX time.

Wall Time: Another name for local time; short for "wall-clock time".

3. The Time Zone Information Format (TZif)

The Time Zone Information Format begins with a fixed 44-octet version 1 header (Section 3.1) containing a field that specifies the version of the file's format. Readers designed for version N can read version N+1 files without too much trouble; data specific to version N+1 either appears after version N data so that earlier version readers can easily ignore later version data they are not designed for, or it appears as a minor extension to version N that version N readers are likely to tolerate well.

The version 1 header is followed by a variable-length version 1 data block (Section 3.2) containing four-octet (32-bit) transition times and leap-second occurrences. These 32-bit values are limited to representing time changes from 1901-12-13 20:45:52 through 2038-01-19 03:14:07 UT, and the version 1 header and data block are present only for backward compatibility with obsolescent readers, as discussed in "Common Interoperability Issues" (Appendix A).

Version 1 files terminate after the version 1 data block. Files from versions 2 and higher extend the format by appending a second 44-octet version 2+ header, a variable-length version 2+ data block containing eight-octet (64-bit) transition times and leap-second occurrences, and a variable-length footer (Section 3.3). These 64-bit values can represent times approximately 292 billion years into the past or future.

NOTE: All multi-octet integer values **MUST** be stored in network octet order format (high-order octet first, otherwise known as big-endian), with all bits significant. Signed integer values **MUST** be represented using two's complement.

A TZif file is structured as follows:

```
   Version 1       Versions 2+
+-------------+   +-------------+
|  Version 1  |   |  Version 1  |
|   Header    |   |   Header    |
+-------------+   +-------------+
|  Version 1  |   |  Version 1  |
|  Data Block |   |  Data Block |
+-------------+   +-------------+
                  |  Version 2+ |
                  |   Header    |
                  +-------------+
                  |  Version 2+ |
                  |  Data Block |
                  +-------------+
                  |   Footer    |
                  +-------------+
```

Figure 1: General Format of TZif Files

3.1. TZif Header

A TZif header is structured as follows (the lengths of multi-octet fields are shown in parentheses):

```
+---------------+---+
|  magic    (4) |ver|
+---------------+---+---------------------------------------+
|           [unused - reserved for future use] (15)         |
+---------------+---------------+---------------+-----------+
|  isutcnt  (4) |  isstdcnt (4) |  leapcnt  (4) |
+---------------+---------------+---------------+
|  timecnt  (4) |  typecnt  (4) |  charcnt  (4) |
+---------------+---------------+---------------+
```

Figure 2: TZif Header

The fields of the header are defined as follows:

magic: The four-octet ASCII [RFC20] sequence "TZif" (0x54 0x5A 0x69 0x66), which identifies the file as utilizing the Time Zone Information Format.

ver(sion): An octet identifying the version of the file's format. The value **MUST** be one of the following:

- NUL (0x00) Version 1 - The file contains only the version 1 header and data block. Version 1 files **MUST NOT** contain a version 2+ header, data block, or footer.
- '2' (0x32) Version 2 - The file **MUST** contain the version 1 header and data block, a version 2+ header and data block, and a footer. The TZ string in the footer (Section 3.3), if non-empty, **MUST** strictly adhere to the requirements for the TZ environment variable as defined in Section 8.3 of the "Base Definitions" volume of [POSIX] and **MUST** encode the POSIX portable character set as ASCII. The leap-second records **MUST NOT** be truncated at the start (Section 6.1) and **MUST NOT** contain an expiration time.

- '3' (0x33) Version 3 - The file **MUST** conform to all version 2 requirements, except that any TZ string in the footer (Section 3.3) **MAY** use the TZ string extension described below (Section 3.3.2).
- '4' (0x34) Version 4 - The file **MUST** conform to all version 3 requirements, except that the leap-second records **MAY** be truncated at the start, and **MAY** contain an expiration time.

isutcnt: A four-octet unsigned integer specifying the number of UT/local indicators contained in the data block -- **MUST** either be zero or equal to "typecnt".

isstdcnt: A four-octet unsigned integer specifying the number of standard/wall indicators contained in the data block -- **MUST** either be zero or equal to "typecnt".

leapcnt: A four-octet unsigned integer specifying the number of leap-second records contained in the data block.

timecnt: A four-octet unsigned integer specifying the number of transition times contained in the data block.

typecnt: A four-octet unsigned integer specifying the number of local time type records contained in the data block -- **MUST NOT** be zero. (Although local time type records convey no useful information in files that have non-empty TZ strings but no transitions, at least one such record is nevertheless required because many TZif readers reject files that have zero time types.)

charcnt: A four-octet unsigned integer specifying the total number of octets used by the set of time zone designations contained in the data block -- **MUST NOT** be zero. The count includes the trailing NUL (0x00) octet at the end of the last time zone designation.

Although the version 1 and 2+ headers have the same format, magic number, and version fields, their count fields may differ, because the version 1 data can be a subset of the version 2+ data.

3.2. TZif Data Block

A TZif data block consists of seven variable-length elements, each of which is a series of items. The number of items in each series is determined by the corresponding count field in the header. The total length of each element is calculated by multiplying the number of items by the size of each item. Therefore, implementations that do not wish to parse or use the version 1 data block can calculate its total length and skip directly to the header of the version 2+ data block.

In the version 1 data block, time values are 32 bits (TIME_SIZE = 4 octets). In the version 2+ data block, present only in version 2 and higher files, time values are 64 bits (TIME_SIZE = 8 octets).

The data block is structured as follows (the lengths of multi-octet fields are shown in parentheses):

```
+---------------------------------------------------------+
|  transition times          (timecnt x TIME_SIZE)        |
+---------------------------------------------------------+
|  transition types          (timecnt)                    |
+---------------------------------------------------------+
|  local time type records   (typecnt x 6)                |
+---------------------------------------------------------+
|  time zone designations    (charcnt)                    |
+---------------------------------------------------------+
|  leap-second records       (leapcnt x (TIME_SIZE + 4))  |
+---------------------------------------------------------+
|  standard/wall indicators  (isstdcnt)                   |
+---------------------------------------------------------+
|  UT/local indicators       (isutcnt)                    |
+---------------------------------------------------------+
```

Figure 3: TZif Data Block

The elements of the data block are defined as follows:

transition times: A series of four- or eight-octet UNIX leap time values sorted in strictly ascending order. Each value is used as a transition time at which the rules for computing local time may change. The number of time values is specified by the "timecnt" field in the header. Each time value **SHOULD** be at least -2^{59}. (-2^{59} is the greatest negated power of 2 that predates the Big Bang, and avoiding earlier timestamps works around known TZif reader bugs relating to outlandishly negative timestamps.)

transition types: A series of one-octet unsigned integers specifying the type of local time of the corresponding transition time. These values serve as zero-based indices into the array of local time type records. The number of type indices is specified by the "timecnt" field in the header. Each type index **MUST** be in the range [0, "typecnt" - 1].

local time type records: A series of six-octet records specifying a local time type. The number of records is specified by the "typecnt" field in the header. Each record has the following format (the lengths of multi-octet fields are shown in parentheses):

```
+---------------+---+---+
|  utoff (4)    |dst|idx|
+---------------+---+---+
```

utoff: A four-octet signed integer specifying the number of seconds to be added to UT in order to determine local time. The value **MUST NOT** be -2^{31} and **SHOULD** be in the range [-89999, 93599] (i.e., its value **SHOULD** be more than -25 hours and less than 26 hours). Avoiding -2^{31} allows 32-bit clients to negate the value without overflow. Restricting it to [-89999, 93599] allows easy support by implementations that already support the POSIX-required range [-24:59:59, 25:59:59].

(is)dst: A one-octet value indicating whether local time should be considered Daylight Saving Time (DST). The value **MUST** be 0 or 1. A value of one (1) indicates that this type of time is DST. A value of zero (0) indicates that this time type is standard time.

(desig)idx: A one-octet unsigned integer specifying a zero-based index into the series of time zone designation octets, thereby selecting a particular designation string. Each index **MUST** be in the range [0, "charcnt" - 1]; it designates the NUL-terminated string of octets starting at position "idx" in the time zone designations. (This string **MAY** be empty.) A NUL octet **MUST** exist in the time zone designations at or after position "idx". If the designation string is "-00", the time type is a placeholder indicating that local time is unspecified.

time zone designations: A series of octets constituting an array of NUL-terminated (0x00) time zone designation strings. The total number of octets is specified by the "charcnt" field in the header. Two designations **MAY** overlap if one is a suffix of the other. The character encoding of time zone designation strings is not specified; however, see Section 4 of this document.

leap-second records: A series of eight- or twelve-octet records specifying the corrections that need to be applied to UTC in order to determine TAI, also known as the leap-second table. The records are sorted by the occurrence time in strictly ascending order. The number of records is specified by the "leapcnt" field in the header. Each record has one of the following structures (the lengths of multi-octet fields are shown in parentheses):

Version 1 Data Block:

```
+---------------+---------------+
|  occur (4)    |  corr (4)     |
+---------------+---------------+
```

version 2+ Data Block:

```
+---------------+---------------+---------------+
|  occur (8)                    |  corr (4)     |
+---------------+---------------+---------------+
```

occur(rence): A four- or eight-octet UNIX leap time value specifying the time at which a leap-second correction occurs or at which the leap-second table expires. The first value, if present, **MUST** be non-negative, and each leap second **MUST** occur at the end of a UTC month.

corr(ection): A four-octet signed integer specifying the value of LEAPCORR on or after the occurrence. If "leapcnt" is zero, LEAPCORR is zero for all timestamps; otherwise, for timestamps before the first occurrence time, LEAPCORR is zero if the first correction is one (1) or minus one (-1) and is unspecified otherwise (which can happen only in files truncated at the start (Section 6.1)).

The first leap second is a positive leap second if and only if its correction is positive. Each correction after the first **MUST** differ from the previous correction by either one (1) for a positive leap second or minus one (-1) for a negative leap second, except that in version 4 files with two or more leap-second records, the correction value of the last two records **MAY** be the same, with the occurrence of last record indicating the expiration time of the leap-second table.

The leap-second table expiration time is the time at which the table no longer records the presence or absence of future leap-second corrections, and post-expiration timestamps cannot be accurately calculated. For example, a leap-second table published in January, which predicts the presence or absence of a leap second at June's end, might expire in mid-December because it is not known when the next leap second will occur.

If leap seconds become permanently discontinued, as requested by the General Conference on Weights and Measures [CGPM-2022-R4], leap-second tables published after the discontinuation time **SHOULD NOT** expire, since they will not be updated in the foreseeable future.

standard/wall indicators: A series of one-octet values indicating whether the transition times associated with local time types were specified as standard time or wall-clock time. Each value **MUST** be 0 or 1. A value of one (1) indicates standard time. The value **MUST** be set to one (1) if the corresponding UT/local indicator is set to one (1). A value of zero (0) indicates wall time. The number of values is specified by the "isstdcnt" field in the header. If "isstdcnt" is zero (0), all transition times associated with local time types are assumed to be specified as wall time.

UT/local indicators: A series of one-octet values indicating whether the transition times associated with local time types were specified as UT or local time. Each value **MUST** be 0 or 1. A value of one (1) indicates UT, and the corresponding standard/wall indicator **MUST** also be set to one (1). A value of zero (0) indicates local time. The number of values is specified by the "isutcnt" field in the header. If "isutcnt" is zero (0), all transition times associated with local time types are assumed to be specified as local time.

The type corresponding to a transition time specifies local time for timestamps starting at the given transition time and continuing up to, but not including, the next transition time. Local time for timestamps before the first transition is specified by the first time type (time type 0). Local time for timestamps on or after the last transition is specified by the TZ string in the footer (Section 3.3) if present and non-empty; otherwise, it is unspecified. If there are no transitions, local time for all timestamps is specified by the TZ string in the footer if present and non-empty; otherwise, it is specified by time type 0. A time type with a designation string of "-00" represents an unspecified local time.

A given pair of standard/wall and UT/local indicators is used to designate whether the corresponding transition time was specified as UT, standard time, or wall-clock time. There are only three combinations of the two indicators, given that the standard/wall value **MUST** be one (1) if the UT/local value is one (1). This information can be useful if the transition times in a TZif file need to be transformed into transitions appropriate for another time zone (e.g., when calculating transition times for a simple POSIX-like TZ string such as "AKST9AKDT").

In order to eliminate unused space in a TZif file, every nonzero local time type index **SHOULD** appear at least once in the transition type array. Likewise, every octet in the time zone designations array **SHOULD** be used by at least one time type record.

3.3. TZif Footer

The TZif footer is structured as follows (the lengths of multi-octet fields are shown in parentheses):

```
+---+--------------------+---+
| NL|  TZ string (0...)  |NL |
+---+--------------------+---+
```

Figure 4: TZif Footer

The elements of the footer are defined as follows:

NL: An ASCII new line character (0x0A).

TZ string: A rule for computing local time changes after the last transition time stored in the version 2+ data block. The string is either empty or uses the expanded format of the "TZ" environment variable as defined in Section 8.3 of the "Base Definitions" volume of [POSIX] with ASCII encoding, possibly utilizing the extension described below (Section 3.3.2) in version 3 and higher files. If the string is empty, the corresponding information is not available. If the string is non-empty and one or more transitions appear in the version 2+ data, the string **MUST** be consistent with the last version 2+ transition. In other words, evaluating the TZ string at the time of the last transition should yield the same time type as was specified in the last transition. The string **MUST NOT** contain NUL octets or be NUL-terminated, and it **SHOULD NOT** begin with the ":" (colon) character.

The TZif footer is present only in version 2 and higher files, as the obsolescent version 1 format was designed before the need for a footer was apparent.

3.3.1. All-Year Daylight Saving Time

DST is considered to be in effect all year if its UT offset is less than (i.e., west of) that of standard time, and it starts January 1 at 00:00 and ends December 31 at 24:00 minus the difference between standard and daylight saving time, leaving no room for standard time in the calendar. [POSIX] implies but does not explicitly state this, so it is spelled out here for clarity.

Example: XXX3EDT4,0/0,J365/23
This represents a time zone that is perpetually 4 hours west of UT and is abbreviated "EDT". The "XXX" is ignored.

3.3.2. TZ String Extension

The TZ string in a version 3 or higher TZif file **MAY** use the following extension to POSIX TZ strings. This extension is described using the terminology of Section 8.3 of the "Base Definitions" volume of [POSIX].

The hours part of the transition times may be signed and range from -167 through 167 (-167 <= hh <= 167) instead of the POSIX-required unsigned values from 0 through 24.

Example: <-03>3<-02>,M3.5.0/-2,M10.5.0/-1
: This represents a time zone that observes daylight saving time from 22:00 on the day before March's last Sunday until 23:00 on the day before October's last Sunday. Standard time is 3 hours west of UT and is abbreviated "-03"; daylight saving time is 2 hours west of UT and is abbreviated "-02".

A TZif file that uses the above extension **MUST** be designated as version 3 (or higher), even if a future version of POSIX adopts this extension.

4. Interoperability Considerations

The following practices help ensure the interoperability of TZif applications.

- Version 1 files are considered a legacy format and **SHOULD NOT** be generated, as they do not support transition times after the year 2038.
- Readers that understand only version 1 **MUST** ignore any data that extends beyond the calculated end of the version 1 data block.
- Other than version 1, writers **SHOULD** generate the lowest version number needed by a file's data. This helps interoperability with older readers. For example, a writer **SHOULD** generate a version 4 file only if its leap-second table either expires or is truncated at the start. Likewise, a writer not generating a version 4 file **SHOULD** generate a version 3 file only if the TZ string extension is necessary to accurately model transition times.
- To save space, writers of version 2+ files **MAY** output a placeholder version 1 data block with all counts zero except that "typecnt" and "charcnt" are both one (1). If this is done, obsolescent version-1-only readers **MUST** interpret these files as lacking time changes and time zone abbreviations.
- Unless the version 1 data block is a placeholder, the sequence of timestamps defined by the version 1 header and data block **SHOULD** be a contiguous sub-sequence of the timestamps defined by the version 2+ header and data block and by the footer. This guideline helps obsolescent version 1 readers agree with current readers about timestamps within the contiguous sub-sequence.
- When a TZif file contains a leap-second table expiration time, TZif readers **SHOULD** either refuse to process post-expiration timestamps or process them as if the expiration time did not exist (possibly with an error indication). This lessens disagreement among implementations when processing far-future timestamps that cannot yet be handled exactly.

- Time zone designations **MUST** consist of at least three (3) and no more than six (6) ASCII characters from the set of alphanumerics, "-", and "+". This is compatible with POSIX requirements for time zone abbreviations.
- A reader that encounters a time zone designation containing bytes other than ASCII alphanumerics, "-", and "+" **SHOULD** act if the designation instead contained a signed numeric string derived from the UT offset, for example, "-10" and "+0530" to indicate 10 hours west and 5.5 hours east of Greenwich, respectively.
- When reading a version 2 or higher file, readers **SHOULD** ignore the version 1 header and data block except for the purpose of skipping over them. This improves compatibility among readers of non-conforming files where version 2+ data is not upward compatible with version 1.
- Readers **SHOULD** calculate the total lengths of the headers and data blocks and check that they all fit within the actual file size, as part of a validity check for the file.
- When a TZif file is used in a MIME message entity, it **SHOULD** be indicated by one of the following media types:
  - "application/tzif-leap" (Section 9.2) to indicate that leap-second records are included in the TZif data as necessary (none are necessary if the file is truncated to a range that precedes the first leap second).
  - "application/tzif" (Section 9.1) to indicate that leap-second records are not included in the TZif data; "leapcnt" in the header(s) **MUST** be zero (0).
- Common interoperability issues and possible workarounds are described in Appendix A.

5. Internationalization Considerations

TZif time zone designations contain only ASCII alphanumerics, "-", and "+". Commonly used designations include numeric strings like "-10" and "+0530" for UT offsets and English language abbreviations like "CEST" for Central European Summer Time and "GMT" for Greenwich Mean Time. It is the TZif reader's responsibility to substitute different abbreviations when needed for internationalization, such as substituting "HNC" (l'heure normale du Centre) for "CST" (Central Standard Time) in French-speaking regions. This substitution can be problematic, as abbreviations can be ambiguous; for example, "CST" commonly stands for China Standard Time and Cuba Standard Time as well as Central Standard Time. One approach for addressing this issue can be found in the time zone charts of the Unicode Common Locale Data Repository (CLDR) Project [CLDR].

Although the original TZif design allowed for any nonzero octets in time zone designations, and it was common practice until the mid-1990s for designations to contain ASCII spaces, designations are now limited to ASCII alphanumerics, "-", and "+" to avoid confusion and to encourage portability to a wide variety of locales.

6. Use with the Time Zone Data Distribution Service

The Time Zone Data Distribution Service (TZDIST) [RFC7808] is a service that allows reliable, secure, and fast delivery of time zone data and leap-second rules to client systems such as calendaring and scheduling applications or operating systems.

A TZDIST service **MAY** supply time zone data to clients in the Time Zone Information Format. Such a service **MUST** indicate that it supports this format by including the media type "application/tzif" (Section 9.1) in its "capabilities" response (Section 5.1 of [RFC7808]). A TZDIST service **MAY** also include the media type "application/tzif-leap" (Section 9.2) in its "capabilities" response if it is able to generate TZif files containing leap-second records. A TZDIST service **MUST NOT** advertise the "application/tzif-leap" media type without also advertising "application/tzif".

TZDIST clients **MUST** use the HTTP "Accept" header field ([RFC9110], Section 12.5.1) to indicate their preference to receive data in the "application/tzif" and/or "application/tzif-leap" formats.

6.1. Truncating TZif Files

As described in Section 3.9 of [RFC7808], a TZDIST service **MAY** truncate time zone transition data. A truncated TZif file is valid from its first and up to, but not including, its last version 2+ transition time, if present.

When truncating the start of a TZif file, the service **MUST** supply in the version 2+ data a first transition time that is the start point of the truncation range. As with untruncated TZif files, time type 0 indicates local time immediately before the start point, and the time type of the first transition indicates local time thereafter. Time type 0 **MUST** be a placeholder indicating that local time is unspecified, so that the reader is unambiguously informed of truncation at the start.

When truncating the start of a TZif file containing leap-second records, the service **MUST** keep all leap-second records governing timestamps within the truncation range, even if the first such record precedes the start point of the truncation range. If the truncated leap-second table is non-empty, its first record **MUST** have a positive correction if and only if it represents a positive leap second.

When truncating the end of a TZif file, the service **MUST** supply in the version 2+ data a last transition time that is the end point of the truncation range and **MUST** supply an empty TZ string. As with untruncated TZif files with empty TZ strings, a truncated TZif file does not indicate local time after the last transition. To this end, the time type of the last transition **MUST** be a placeholder indicating that local time is unspecified.

All represented information that falls inside the truncation range **MUST** be the same as that represented by a corresponding untruncated TZif file.

TZDIST clients **SHOULD NOT** use a truncated TZif file (as described above) to interpret timestamps outside the truncation time range.

6.2. Example TZDIST Request for TZif Data

In this example, the client checks the server for the available formats and then requests that the time zone with a specific time zone identifier be returned in Time Zone Information Format.

This example presumes that the time zone context path has been discovered (see [RFC7808], Section 4.2.1) to be "/tzdist".

```
>> Request <<

GET /tzdist/capabilities HTTP/1.1
Host: tz.example.com

>> Response <<

HTTP/1.1 200 OK
Date: Fri, 01 Jun 2018 14:52:23 GMT
Content-Type: application/json
Content-Length: xxxx

{
  "version": 1,

  "info": {
    "primary-source": "IANA:2018e",
    "formats": [
      "text/calendar",
      "application/tzif",
      "application/tzif-leap"
    ],
...
  },
...
}


>> Request <<

GET /tzdist/zones/America%2FNew_York HTTP/1.1
Host: tz.example.com
Accept: application/tzif

>> Response <<

HTTP/1.1 200 OK
Date: Fri, 01 Jun 2018 14:52:24 GMT
Content-Type: application/tzif
Content-Length: xxxx
ETag: "123456789-000-111"

TZif2...[binary data without leap-second records]...
EST5EDT,M3.2.0,M11.1.0
```

7. Security Considerations

The Time Zone Information Format contains no executable code, and it does not define any extensible areas that could be used to store such code.

TZif contains counted arrays of data elements. All counts should be checked when processing TZif objects, to guard against references past the end of the object.

TZif provides no confidentiality or integrity protection. Time zone information is normally public and does not call for confidentiality protection. Since time zone information is used in many critical applications, integrity protection may be required and must be provided externally.

As discussed in Section 8 of [RFC7808], transmission of time zone data over an insecure communication channel could result in tampered data, harming calendaring and scheduling operations. As such, TZif data transmitted over a public communications channel **MUST** be protected with a security layer such as that provided by Transport Layer Security (TLS) [RFC8446].

8. Privacy Considerations

The Time Zone Information Format contains publicly available data, and it does not define any extensible areas that could be used to store private data.

As discussed in Section 9 of [RFC7808], transmission of time zone data over an insecure communications channel could leak the past, current, or future location of a device or user. As such, TZif data transmitted over a public communications channel **MUST** be protected with a confidentiality layer such as that provided by Transport Layer Security (TLS) [RFC8446].

9. IANA Considerations

IANA has updated the "Media Types" registry as follows.

This document defines two media types [RFC6838] for the exchange of data utilizing the Time Zone Information Format.

9.1. application/tzif

Type name: application

Subtype name: tzif

Required parameters: N/A

Optional parameters: N/A

Encoding considerations: binary

Security considerations: See Section 7 of RFC 9636.

Interoperability considerations: See Section 4 of RFC 9636.

Published specification: RFC 9636.

Applications that use this media type: This media type is designed for widespread use by applications that need to use or exchange time zone information relative to UNIX time, such as the Time Zone Information Compiler (zic) [ZIC] and the GNU C Library [GNU-C]. The Time Zone Distribution Service [RFC7808] can directly use this media type.

Fragment identifier considerations: N/A

Additional information:

Magic number(s): The first 4 octets are 0x54, 0x5A, 0x69, 0x66
File extensions(s): N/A
Macintosh file type code(s): N/A

Person & email address to contact for further information: Time Zone Database mailing list <tz@iana.org>

Intended usage: COMMON

Restrictions on usage: N/A

Author: See the "Authors' Addresses" section of RFC 9636.

Change controller: IETF

9.2. application/tzif-leap

Type name: application

Subtype name: tzif-leap

Required parameters: none

Optional parameters: none

Encoding considerations: binary

Security considerations: See Section 7 of RFC 9636.

Interoperability considerations: See Section 4 of RFC 9636.

Published specification: RFC 9636.

Applications that use this media type: This media type is designed for widespread use by applications that need to use or exchange time zone information relative to UNIX leap time, such as the Time Zone Information Compiler (zic) [ZIC] and the GNU C Library [GNU-C]. The Time Zone Distribution Service [RFC7808] can directly use this media type.

Fragment identifier considerations: N/A

Additional information:

Magic number(s): The first 4 octets are 0x54, 0x5A, 0x69, 0x66
File extensions(s): N/A
Macintosh file type code(s): N/A

Person & email address to contact for further information: Time Zone Database mailing list <tz@iana.org>

Intended usage: COMMON

Restrictions on usage: N/A

Author: See the "Authors' Addresses" section of RFC 9636.

Change controller: IETF

10. References

10.1. Normative References

[GNU-C] Free Software Foundation, "The GNU C Library", <https://www.gnu.org/software/libc/>.

[ITU-R-TF.460] International Telecommunication Union, "Standard-frequency and time-signal emissions", ITU-R Recommendation TF.460, February 2002, <https://www.itu.int/rec/R-REC-TF.460/en>.

[POSIX] IEEE, "IEEE Standard for Information Technology--Portable Operating System Interface (POSIX(TM)) Base Specifications, Issue 7", POSIX.1-2017, IEEE Std 1003.1-2017, DOI 10.1109/IEEESTD.2018.8277153, January 2018, <https://pubs.opengroup.org/onlinepubs/9699919799/>.

[RFC20] Cerf, V., "ASCII format for network interchange", STD 80, RFC 20, DOI 10.17487/RFC0020, October 1969, <https://www.rfc-editor.org/info/rfc20>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <https://www.rfc-editor.org/info/rfc6838>.

[RFC7808] Douglass, M. and C. Daboo, "Time Zone Data Distribution Service", RFC 7808, DOI 10.17487/RFC7808, March 2016, <https://www.rfc-editor.org/info/rfc7808>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC9110] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Semantics", STD 97, RFC 9110, DOI 10.17487/RFC9110, June 2022, <https://www.rfc-editor.org/info/rfc9110>.

[ZIC] Kerrisk, M., "zic(8) - Linux manual page", <http://man7.org/linux/man-pages/man8/zic.8.html>.

10.2. Informative References

[CGPM-2022-R4] General Conference on Weights and Measures, "Resolution 4 of the 27th CGPM (2022)", DOI 10.59161/CGPM2022RES4E, November 2022, <https://www.bipm.org/en/cgpm-2022/resolution-4>.

[CLDR] Unicode, Inc., "Unicode CLDR Project", <https://cldr.unicode.org/>.

[EGGERT-TZ] "History for tz", commit b5318b5, June 2024, <https://github.com/eggert/tz/commits/main/tzfile.5>.

[Err6426] RFC Errata, "Erratum ID 6426", RFC 8536, <https://www.rfc-editor.org/errata/eid6426>.

[Err6435] RFC Errata, "Erratum ID 6435", RFC 8536, <https://www.rfc-editor.org/errata/eid6435>.

[Err6757] RFC Errata, "Erratum ID 6757", RFC 8536, <https://www.rfc-editor.org/errata/eid6757>.

[RFC5545] Desruisseaux, B., Ed., "Internet Calendaring and Scheduling Core Object Specification (iCalendar)", RFC 5545, DOI 10.17487/RFC5545, September 2009, <https://www.rfc-editor.org/info/rfc5545>.

[RFC6557] Lear, E. and P. Eggert, "Procedures for Maintaining the Time Zone Database", BCP 175, RFC 6557, DOI 10.17487/RFC6557, February 2012, <https://www.rfc-editor.org/info/rfc6557>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC8536] Olson, A., Eggert, P., and K. Murchison, "The Time Zone Information Format (TZif)", RFC 8536, DOI 10.17487/RFC8536, February 2019, <https://www.rfc-editor.org/info/rfc8536>.

[tz-link] Eggert, P. and A. Olson, "Time zone and daylight saving time data", <https://www.iana.org/time-zones/repository/tz-link.html>.

Appendix A. Common Interoperability Issues

This section documents common problems in implementing this specification. Most of these are problems in generating TZif files for use by readers conforming to predecessors of this specification [EGGERT-TZ]. The goals of this section are to help:

1. TZif writers output files that avoid common pitfalls in older or buggy TZif readers,
2. TZif readers avoid common pitfalls when reading files generated by future TZif writers, and
3. any future specification authors see what sort of problems arise when the TZif format is changed.

When new versions of the TZif format have been defined, a design goal has been that a reader can successfully use a TZif file even if the file is of a later TZif version than what the reader was designed for. When complete compatibility was not achieved, an attempt was made to limit glitches to rarely used timestamps and allow simple partial workarounds in writers designed to generate newer-version data useful even for older-version readers. This section attempts to document these compatibility issues and workarounds as well as other common bugs in readers.

Interoperability problems with TZif include the following:

- Some readers examine only version 1 data. As a partial workaround, a writer can output as much version 1 data as possible. However, a reader should ignore version 1 data and use version 2+ data, even if the reader's timestamps have only 32 bits.
- Some readers designed for version 2 might mishandle timestamps after a version 3 or higher file's last transition, because they cannot parse extensions to POSIX in the TZ-like string. As a partial workaround, a writer can output more transitions than necessary, so that only far-future timestamps are mishandled by version 2 readers.
- Some readers designed for version 2 do not support permanent daylight saving time with transitions after 24:00 -- e.g., a TZ string "EST5EDT,0/0,J365/25" denoting permanent Eastern Daylight Time (-04). As a workaround, a writer can substitute standard time for two time zones east, e.g., "XXX3EDT4,0/0,J365/23" for a time zone with a never-used standard time (XXX, -03) and negative daylight saving time (EDT, -04) all year. Alternatively, as a partial workaround, a writer can substitute standard time for the next time zone east -- e.g., "AST4" for permanent Atlantic Standard Time (-04).
- Some readers designed for version 2 or 3 and that require strict conformance to [RFC8536] reject version 4 files whose leap-second tables are truncated at the start or end in expiration times.
- Some readers ignore the footer and instead predict future timestamps from the time type of the last transition. As a partial workaround, a writer can output more transitions than necessary.
- Some readers do not use time type 0 for timestamps before the first transition, in that they infer a time type using a heuristic that does not always select time type 0. As a partial workaround, a writer can output a placeholder (no-op) first transition at an early time.

- Some readers mishandle timestamps before the first transition that has a timestamp that is not less than -2^{31}. Readers that support only 32-bit timestamps are likely to be more prone to this problem, for example, when they process 64-bit transitions, only some of which are representable in 32 bits. As a partial workaround, a writer can output a placeholder transition at timestamp -2^{31}.
- Some readers mishandle a transition if its timestamp has the minimum possible signed 64-bit value. Timestamps less than -2^{59} are not recommended.
- Some readers mishandle POSIX-style TZ strings that contain "<" or ">". As a partial workaround, a writer can avoid using "<" or ">" for time zone abbreviations containing only alphabetic characters.
- Many readers mishandle time zone abbreviations that contain non-ASCII characters. These characters are not recommended.
- Some readers may mishandle time zone abbreviations that contain fewer than 3 or more than 6 characters or that contain ASCII characters other than alphanumerics, "-", and "+". These abbreviations are not recommended.
- This specification does not dictate how readers should deal with timestamps when local time is unspecified. Common practice is for readers to report UT with designation string "-00". A reader could return an error indication instead.
- Some readers mishandle TZif files that specify daylight saving time UT offsets that are less than the UT offsets for the corresponding standard time. These readers do not support locations like Ireland, which uses the equivalent of the POSIX TZ string "IST-1GMT0,M10.5.0,M3.5.0/1", observing standard time (IST, +01) in summer and daylight saving time (GMT, +00) in winter. As a partial workaround, a writer can output data for the equivalent of the POSIX TZ string "GMT0IST,M3.5.0/1,M10.5.0", thus swapping standard and daylight saving time. Although this workaround misidentifies which part of the year uses daylight saving time, it records UT offsets and time zone abbreviations correctly.
- Some readers generate ambiguous timestamps for positive leap seconds that occur when the UTC offset is not a multiple of 60 seconds. For example, in a time zone with UTC offset +01:23:45 and with a positive leap second 78796801 (1972-06-30 23:59:60 UTC), some readers will map both 78796800 and 78796801 to 01:23:45 local time the next day instead of mapping the latter to 01:23:46, and they will map 78796815 to 01:23:59 instead of to 01:23:60. This has not yet been a practical problem, since no civil authority has observed such UTC offsets since leap seconds were introduced in 1972.

Some interoperability problems are reader bugs that are listed here mostly as warnings to developers of readers.

- Some readers do not support negative timestamps. Developers of distributed applications should keep this in mind if they need to deal with pre-1970 data.
- Some readers mishandle timestamps before the first transition that has a non-negative timestamp. Readers that do not support negative timestamps are likely to be more prone to this problem.
- Some readers mishandle time zone abbreviations like "-08" that contain "+", "-", or digits.

- Some readers mishandle UT offsets that are out of the conventional range of -12 through +12 hours and so do not support locations like Kiritimati that are outside this range.
- Some readers mishandle UT offsets in the range [-3599, -1] seconds from UT because they integer-divide the offset by 3600 to get 0 and then display the hour part as "+00".
- Some readers mishandle UT offsets that are not a multiple of one hour, 15 minutes, or 1 minute.

Appendix B. Example TZif Files

The following sections contain annotated hexadecimal dumps of example TZif files.

These examples should only be considered informative. Although the example data entries are current as of the publication date of this document, the data will likely change in the future as leap seconds are added and changes are made to civil time.

B.1. Version 1 File Representing UTC (with Leap Seconds)

File Offset	Hexadecimal Octets	Record Name / Field Name	Field Value
000	54 5a 69 66	magic	"TZif"
004	00	version	0 (1)
005	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		
020	00 00 00 01	isutcnt	1
024	00 00 00 01	isstdcnt	1
028	00 00 00 1b	leapcnt	27
032	00 00 00 00	timecnt	0
036	00 00 00 01	typecnt	1
040	00 00 00 04	charcnt	4
		localtimetype[0]	
044	00 00 00 00	utoff	0 (+00:00)
048	00	isdst	0 (no)
049	00	desigidx	0
050	55 54 43 00	designations[0]	"UTC\0"

File Offset	Hexadecimal Octets	Record Name / Field Name	Field Value
		leapsecond[0]	
054	04 b2 58 00	occurrence	78796800 (1972-06-30T23:59:60Z)
058	00 00 00 01	correction	1
		leapsecond[1]	
062	05 a4 ec 01	occurrence	94694401 (1972-12-31T23:59:60Z)
066	00 00 00 02	correction	2
		leapsecond[2]	
070	07 86 1f 82	occurrence	126230402 (1973-12-31T23:59:60Z)
074	00 00 00 03	correction	3
		leapsecond[3]	
078	09 67 53 03	occurrence	157766403 (1974-12-31T23:59:60Z)
082	00 00 00 04	correction	4
		leapsecond[4]	
086	0b 48 86 84	occurrence	189302404 (1975-12-31T23:59:60Z)
090	00 00 00 05	correction	5
		leapsecond[5]	
094	0d 2b 0b 85	occurrence	220924805 (1976-12-31T23:59:60Z)
098	00 00 00 06	correction	6
		leapsecond[6]	
102	0f 0c 3f 06	occurrence	252460806 (1977-12-31T23:59:60Z)

File Offset	Hexadecimal Octets	Record Name / Field Name	Field Value
106	00 00 00 07	correction	7
		leapsecond[7]	
110	10 ed 72 87	occurrence	283996807 (1978-12-31T23:59:60Z)
114	00 00 00 08	correction	8
		leapsecond[8]	
118	12 ce a6 08	occurrence	315532808 (1979-12-31T23:59:60Z)
122	00 00 00 09	correction	9
		leapsecond[9]	
126	15 9f ca 89	occurrence	362793609 (1981-06-30T23:59:60Z)
130	00 00 00 0a	correction	10
		leapsecond[10]	
134	17 80 fe 0a	occurrence	394329610 (1982-06-30T23:59:60Z)
138	00 00 00 0b	correction	11
		leapsecond[11]	
142	19 62 31 8b	occurrence	425865611 (1983-06-30T23:59:60Z)
146	00 00 00 0c	correction	12
		leapsecond[12]	
150	1d 25 ea 0c	occurrence	489024012 (1985-06-30T23:59:60Z)
154	00 00 00 0d	correction	13
		leapsecond[13]	

File Offset	Hexadecimal Octets	Record Name / Field Name	Field Value
158	21 da e5 0d	occurrence	567993613 (1987-12-31T23:59:60Z)
162	00 00 00 0e	correction	14
		leapsecond[14]	
166	25 9e 9d 8e	occurrence	631152014 (1989-12-31T23:59:60Z)
170	00 00 00 0f	correction	15
		leapsecond[15]	
174	27 7f d1 0f	occurrence	662688015 (1990-12-31T23:59:60Z)
178	00 00 00 10	correction	16
		leapsecond[16]	
182	2a 50 f5 90	occurrence	709948816 (1992-06-30T23:59:60Z)
186	00 00 00 11	correction	17
		leapsecond[17]	
190	2c 32 29 11	occurrence	741484817 (1993-06-30T23:59:60Z)
194	00 00 00 12	correction	18
		leapsecond[18]	
198	2e 13 5c 92	occurrence	773020818 (1994-06-30T23:59:60Z)
202	00 00 00 13	correction	19
		leapsecond[19]	
206	30 e7 24 13	occurrence	820454419 (1995-12-31T23:59:60Z)
210	00 00 00 14	correction	20

File Offset	Hexadecimal Octets	Record Name / Field Name	Field Value
		leapsecond[20]	
214	33 b8 48 94	occurrence	867715220 (1997-06-30T23:59:60Z)
218	00 00 00 15	correction	21
		leapsecond[21]	
222	36 8c 10 15	occurrence	915148821 (1998-12-31T23:59:60Z)
226	00 00 00 16	correction	22
		leapsecond[22]	
230	43 b7 1b 96	occurrence	1136073622 (2005-12-31T23:59:60Z)
234	00 00 00 17	correction	23
		leapsecond[23]	
238	49 5c 07 97	occurrence	1230768023 (2008-12-31T23:59:60Z)
242	00 00 00 18	correction	24
		leapsecond[24]	
246	4f ef 93 18	occurrence	1341100824 (2012-06-30T23:59:60Z)
250	00 00 00 19	correction	25
		leapsecond[25]	
254	55 93 2d 99	occurrence	1435708825 (2015-06-30T23:59:60Z)
258	00 00 00 1a	correction	26
		leapsecond[26]	
262	58 68 46 9a	occurrence	1483228826 (2016-12-31T23:59:60Z)

File Offset	Hexadecimal Octets	Record Name / Field Name	Field Value
266	00 00 00 1b	correction	27
270	00	standard/wall[0]	0 (wall)
271	00	UT/local[0]	0 (local)

Table 1

To determine TAI corresponding to 2000-01-01T00:00:00Z (UNIX time = 946684800), the following procedure would be followed:

1. Find the latest leap-second occurrence prior to the time of interest (leapsecond[21]) and note the correction value (LEAPCORR = 22).
2. Add LEAPCORR + 10 to the time of interest to yield TAI of 2000-01-01T00:00:32.

B.2. Version 2 File Representing Pacific/Honolulu

File Offset	Hexadecimal Octets	Record Name / Field Name	Field Value
000	54 5a 69 66	magic	"TZif"
004	32	version	'2' (2)
005	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		
020	00 00 00 06	isutcnt	6
024	00 00 00 06	isstdcnt	6
028	00 00 00 00	leapcnt	0
032	00 00 00 07	timecnt	7
036	00 00 00 06	typecnt	6
040	00 00 00 14	charcnt	20
044	80 00 00 00	trans time[0]	-2147483648 (1901-12-13T20:45:52Z)
048	bb 05 43 48	trans time[1]	-1157283000 (1933-04-30T12:30:00Z)

File Offset	Hexadecimal Octets	Record Name / Field Name	Field Value
052	bb 21 71 58	trans time[2]	-1155436200 (1933-05-21T21:30:00Z)
056	cb 89 3d c8	trans time[3]	-880198200 (1942-02-09T12:30:00Z)
060	d2 23 f4 70	trans time[4]	-769395600 (1945-08-14T23:00:00Z)
064	d2 61 49 38	trans time[5]	-765376200 (1945-09-30T11:30:00Z)
068	d5 8d 73 48	trans time[6]	-712150200 (1947-06-08T12:30:00Z)
072	01	trans type[0]	1
073	02	trans type[1]	2
074	01	trans type[2]	1
075	03	trans type[3]	3
076	04	trans type[4]	4
077	01	trans type[5]	1
078	05	trans type[6]	5
		localtimetype[0]	
079	ff ff 6c 02	utoff	-37886 (-10:31:26)
083	00	isdst	0 (no)
084	00	desigidx	0
		localtimetype[1]	
085	ff ff 6c 58	utoff	-37800 (-10:30)
089	00	isdst	0 (no)
090	04	desigidx	4
		localtimetype[2]	

File Offset	Hexadecimal Octets	Record Name / Field Name	Field Value
091	ff ff 7a 68	utoff	-34200 (-09:30)
095	01	isdst	1 (yes)
096	08	desigidx	8
		localtimetype[3]	
097	ff ff 7a 68	utoff	-34200 (-09:30)
101	01	isdst	1 (yes)
102	0c	desigidx	12
		localtimetype[4]	
103	ff ff 7a 68	utoff	-34200 (-09:30)
107	01	isdst	1 (yes)
108	10	desigidx	16
		localtimetype[5]	
109	ff ff 73 60	utoff	-36000 (-10:00)
113	00	isdst	0 (no)
114	04	desigidx	4
115	4c 4d 54 00	designations[0]	"LMT\0"
119	48 53 54 00	designations[4]	"HST\0"
123	48 44 54 00	designations[8]	"HDT\0"
127	48 57 54 00	designations[12]	"HWT\0"
131	48 50 54 00	designations[16]	"HPT\0"
135	00	standard/wall[0]	0 (wall)
136	00	standard/wall[1]	0 (wall)
137	00	standard/wall[2]	0 (wall)
138	00	standard/wall[3]	0 (wall)

File Offset	Hexadecimal Octets	Record Name / Field Name	Field Value
139	01	standard/wall[4]	1 (standard)
140	00	standard/wall[5]	0 (wall)
141	00	UT/local[0]	0 (local)
142	00	UT/local[1]	0 (local)
143	00	UT/local[2]	0 (local)
144	00	UT/local[3]	0 (local)
145	01	UT/local[4]	1 (UT)
146	00	UT/local[5]	0 (local)
147	54 5a 69 66	magic	"TZif"
151	32	version	'2' (2)
152	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		
167	00 00 00 06	isutcnt	6
171	00 00 00 06	isstdcnt	6
175	00 00 00 00	leapcnt	0
179	00 00 00 07	timecnt	7
183	00 00 00 06	typecnt	6
187	00 00 00 14	charcnt	20
191	ff ff ff ff 74 e0 70 be	trans time[0]	-2334101314 (1896-01-13T22:31:26Z)
199	ff ff ff ff bb 05 43 48	trans time[1]	-1157283000 (1933-04-30T12:30:00Z)
207	ff ff ff ff bb 21 71 58	trans time[2]	-1155436200 (1933-05-21T21:30:00Z)
215	ff ff ff ff cb 89 3d c8	trans time[3]	-880198200 (1942-02-09T12:30:00Z)

File Offset	Hexadecimal Octets	Record Name / Field Name	Field Value
223	ff ff ff ff d2 23 f4 70	trans time[4]	-769395600 (1945-08-14T23:00:00Z)
231	ff ff ff ff d2 61 49 38	trans time[5]	-765376200 (1945-09-30T11:30:00Z)
239	ff ff ff ff d5 8d 73 48	trans time[6]	-712150200 (1947-06-08T12:30:00Z)
247	01	trans type[0]	1
248	02	trans type[1]	2
249	01	trans type[2]	1
250	03	trans type[3]	3
251	04	trans type[4]	4
252	01	trans type[5]	1
253	05	trans type[6]	5
		localtimetype[0]	
254	ff ff 6c 02	utoff	-37886 (-10:31:26)
258	00	isdst	0 (no)
259	00	desigidx	0
		localtimetype[1]	
260	ff ff 6c 58	utoff	-37800 (-10:30)
264	00	isdst	0 (no)
265	04	desigidx	4
		localtimetype[2]	
266	ff ff 7a 68	utoff	-34200 (-09:30)
270	01	isdst	1 (yes)
271	08	desigidx	8

File Offset	Hexadecimal Octets	Record Name / Field Name	Field Value
		localtimetype[3]	
272	ff ff 7a 68	utoff	-34200 (-09:30)
276	01	isdst	1 (yes)
277	0c	desigidx	12
		localtimetype[4]	
278	ff ff 7a 68	utoff	-34200 (-09:30)
282	01	isdst	1 (yes)
283	10	desigidx	16
		localtimetype[5]	
284	ff ff 73 60	utoff	-36000 (-10:00)
288	00	isdst	0 (no)
289	04	desigidx	4
290	4c 4d 54 00	designations[0]	"LMT\0"
294	48 53 54 00	designations[4]	"HST\0"
298	48 44 54 00	designations[8]	"HDT\0"
302	48 57 54 00	designations[12]	"HWT\0"
306	48 50 54 00	designations[16]	"HPT\0"
310	00	standard/wall[0]	0 (wall)
311	00	standard/wall[1]	0 (wall)
312	00	standard/wall[2]	0 (wall)
313	00	standard/wall[3]	0 (wall)
314	01	standard/wall[4]	1 (standard)
315	00	standard/wall[5]	0 (wall)
316	00	UT/local[0]	0 (local)

File Offset	Hexadecimal Octets	Record Name / Field Name	Field Value
317	00	UT/local[1]	0 (local)
318	00	UT/local[2]	0 (local)
319	00	UT/local[3]	0 (local)
320	01	UT/local[4]	1 (UT)
321	00	UT/local[5]	0 (local)
322	0a	NL	'\n'
323	48 53 54 31 30	TZ string	"HST10"
328	0a	NL	'\n'

Table 2

To determine the local time in this time zone corresponding to 1933-05-04T12:00:00Z (UNIX time = -1156939200), the following procedure would be followed:

1. Find the latest time transition prior to the time of interest (trans time[1]).
2. Reference the corresponding transition type (trans type[1]) to determine the local time type index (2).
3. Reference the corresponding local time type (localtimetype[2]) to determine the offset from UTC (-09:30), the daylight saving indicator (1 = yes), and the index into the time zone designation strings (8).
4. Look up the corresponding time zone designation string (designations[8] = "HDT").
5. Add the UTC offset to the time of interest to yield a local daylight saving time of 1933-05-04T02:30:00-09:30 (HDT).

To determine the local time in this time zone corresponding to 2019-01-01T00:00:00Z (UNIX time = 1546300800), the following procedure would be followed:

1. Find the latest time transition prior to the time of interest (there is no such transition).
2. Look up the TZ string in the footer ("HST10"), which indicates that the time zone designation is "HST" year-round and the offset to UTC is 10:00.
3. Subtract the UTC offset from the time of interest to yield a standard local time of 2018-12-31T14:00:00-10:00 (HST).

B.3. Truncated Version 2 File Representing Pacific/Johnston

The following TZif file has been truncated to end on 2004-06-161T00:00:00Z (the atoll was abandoned sometime on 2004-06-15).

In this example:

- The version 1 header contains only the required minimum data, which will be ignored by readers.
- The version 2 header leverages the fact that, by specifying 'isutcnt' and 'isstdcnt' as zero, all transition times associated with local time types are assumed to be specified as local wall-clock time (see the definitions of UT/local indicators and standard/wall indicators in Section 3.2).
- The time type of the last transition has designation "-00", indicating that local time is unspecified.
- The TZ string is empty, indicating that there are no known future transitions.

File Offset	Hexadecimal Octets	Record Name / Field Name	Field Value
000	54 5a 69 66	magic	"TZif"
004	32	version	'2' (2)
005	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		
020	00 00 00 00	isutcnt	0
024	00 00 00 00	isstdcnt	0
028	00 00 00 00	leapcnt	0
032	00 00 00 00	timecnt	0
036	00 00 00 01	typecnt	1
040	00 00 00 01	charcnt	1
		localtimetype[0]	
044	00 00 00 00	utoff	0 (+00:00)
048	00	isdst	0 (no)
049	00	desigidx	0
050	00	designations[0]	"\0"
051	54 5a 69 66	magic	"TZif"
055	32	version	'2' (2)

File Offset	Hexadecimal Octets	Record Name / Field Name	Field Value
056	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		
071	00 00 00 00	isutcnt	0
075	00 00 00 00	isstdcnt	0
079	00 00 00 00	leapcnt	0
083	00 00 00 08	timecnt	8
087	00 00 00 07	typecnt	7
091	00 00 00 18	charcnt	24
095	ff ff ff ff 74 e0 70 be	trans time[0]	-2334101314 (1896-01-13T22:31:26Z)
103	ff ff ff ff bb 05 43 48	trans time[1]	-1157283000 (1933-04-30T12:30:00Z)
111	ff ff ff ff bb 21 71 58	trans time[2]	-1155436200 (1933-05-21T21:30:00Z)
119	ff ff ff ff cb 89 3d c8	trans time[3]	-880198200 (1942-02-09T12:30:00Z)
127	ff ff ff ff d2 23 f4 70	trans time[4]	-769395600 (1945-08-14T23:00:00Z)
135	ff ff ff ff d2 61 49 38	trans time[5]	-765376200 (1945-09-30T11:30:00Z)
143	ff ff ff ff d5 8d 73 48	trans time[6]	-712150200 (1947-06-08T12:30:00Z)
151	00 00 00 00 40 cf 8d 80	trans time[7]	1087344000 (2004-06-16T00:00:00Z)
159	02	trans type[0]	2
160	03	trans type[1]	3
161	02	trans type[2]	2
162	04	trans type[3]	4

File Offset	Hexadecimal Octets	Record Name / Field Name	Field Value
163	05	trans type[4]	5
164	02	trans type[5]	2
165	06	trans type[6]	6
166	01	trans type[7]	1
		localtimetype[0]	
167	ff ff 6c 02	utoff	-37886 (-10:31:26)
171	00	isdst	0 (no)
172	04	desigidx	4
		localtimetype[1]	
173	00 00 00 00	utoff	0 (+00:00)
177	00	isdst	0 (no)
178	00	desigidx	0
		localtimetype[2]	
179	ff ff 6c 58	utoff	-37800 (-10:30)
183	00	isdst	0 (no)
184	08	desigidx	8
		localtimetype[3]	
185	ff ff 7a 68	utoff	-34200 (-09:30)
189	01	isdst	1 (yes)
190	0c	desigidx	12
		localtimetype[4]	
191	ff ff 7a 68	utoff	-34200 (-09:30)
195	01	isdst	1 (yes)
196	10	desigidx	16

File Offset	Hexadecimal Octets	Record Name / Field Name	Field Value
		localtimetype[5]	
197	ff ff 7a 68	utoff	-34200 (-09:30)
201	01	isdst	1 (yes)
202	14	desigidx	20
		localtimetype[6]	
203	ff ff 73 60	utoff	-36000 (-10:00)
207	00	isdst	0 (no)
208	08	desigidx	8
209	2d 30 30 00	designations[0]	"-00\0"
213	4c 4d 54 00	designations[4]	"LMT\0"
217	48 53 54 00	designations[8]	"HST\0"
221	48 44 54 00	designations[12]	"HDT\0"
225	48 57 54 00	designations[16]	"HWT\0"
229	48 50 54 00	designations[20]	"HPT\0"
233	0a	NL	'\n'
234		TZ string	""
234	0a	NL	'\n'

Table 3

B.4. Truncated Version 3 File Representing Asia/Jerusalem

The following TZif file has been truncated to start on 2038-01-01T00:00:00Z.

In this example:

- The start time value cannot be represented using 32 bits, so the version 1 header contains only the required minimum data, which will be ignored by readers.
- The version 3 header leverages the fact that, by specifying 'isutcnt' and 'isstdcnt' as zero, all transition times associated with local time types are assumed to be specified as local wall-

clock time (see the definitions of UT/local indicators and standard/wall indicators in Section 3.2).

- Time type 0 has designation "-00", indicating that local time is unspecified prior to the truncation time.
- The TZ string value has been line-wrapped for presentation purposes only.

File Offset	Hexadecimal Octets	Record Name / Field Name	Field Value
000	54 5a 69 66	magic	"TZif"
004	33	version	'3' (3)
005	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		
020	00 00 00 00	isutcnt	0
024	00 00 00 00	isstdcnt	0
028	00 00 00 00	leapcnt	0
032	00 00 00 00	timecnt	0
036	00 00 00 01	typecnt	1
040	00 00 00 01	charcnt	1
		localtimetype[0]	
044	00 00 00 00	utoff	0 (+00:00)
048	00	isdst	0 (no)
049	00	desigidx	0
050	00	designations[0]	"\0"
051	54 5a 69 66	magic	"TZif"
055	33	version	'3' (3)
056	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		
071	00 00 00 00	isutcnt	0
075	00 00 00 00	isstdcnt	0

File Offset	Hexadecimal Octets	Record Name / Field Name	Field Value
079	00 00 00 00	leapcnt	0
083	00 00 00 01	timecnt	1
087	00 00 00 02	typecnt	2
091	00 00 00 08	charcnt	8
095	00 00 00 00 7f e8 17 80	trans time[0]	2145916800 (2038-01-01T00:00:00Z)
103	01	trans type[0]	1
		localtimetype[0]	
104	00 00 00 00	utoff	0 (+00:00)
108	00	isdst	0 (no)
109	00	desigidx	0
		localtimetype[1]	
110	00 00 1c 20	utoff	7200 (+02:00)
114	00	isdst	0 (no)
115	04	desigidx	4
116	2d 30 30 00	designations[0]	"-00\0"
120	49 53 54 00	designations[4]	"IST\0"
124	0a	NL	'\n'
125	49 53 54 2d 32 49 44 54 2c 4d 33 2e 34 2e 34 2f 32 36 2c 4d 31 30 2e 35 2e 30	TZ string	"IST-2IDT,M3.4.4/26,M10.5.0"
151	0a	NL	'\n'

Table 4

B.5. Truncated Version 4 File Representing Europe/London

The following TZif file has been truncated to start on 2022-01-01T00:00:00Z.

In this example:

- The version 1 header contains only the required minimum data, which will be ignored by readers.
- The version 4 header leverages the fact that, by specifying 'isutcnt' and 'isstdcnt' as zero, all transition times associated with local time types are assumed to be specified as local wall-clock time (see the definitions of UT/local indicators and standard/wall indicators in Section 3.2).
- Time type 0 has designation "-00", indicating that local time is unspecified prior to the truncation time.
- The first leap-second occurrence is the most recent one prior to the truncation time.
- The last leap-second correction matches the second-to-last leap-second correction, indicating the expiration time of the leap-second table.
- The TZ string value has been line-wrapped for presentation purposes only.

File Offset	Hexadecimal Octets	Record Name / Field Name	Field Value
000	54 5a 69 66	magic	"TZif"
004	34	version	'4' (4)
005	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		
020	00 00 00 00	isutcnt	0
024	00 00 00 00	isstdcnt	0
028	00 00 00 00	leapcnt	0
032	00 00 00 00	timecnt	0
036	00 00 00 01	typecnt	1
040	00 00 00 01	charcnt	1
		localtimetype[0]	
044	00 00 00 00	utoff	0 (+00:00)
048	00	isdst	0 (no)
049	00	desigidx	0
050	00	designations[0]	"\0"

File Offset	Hexadecimal Octets	Record Name / Field Name	Field Value
051	54 5a 69 66	magic	"TZif"
055	34	version	'4' (4)
056	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		
071	00 00 00 00	isutcnt	0
075	00 00 00 00	isstdcnt	0
079	00 00 00 02	leapcnt	2
083	00 00 00 01	timecnt	1
087	00 00 00 02	typecnt	2
091	00 00 00 08	charcnt	8
095	00 00 00 00 61 cf 99 9b	trans time[0]	1640995227 (2022-01-01T00:00:27Z)
103	01	trans type[0]	1
		localtimetype[0]	
104	00 00 00 00	utoff	0 (+00:00)
108	00	isdst	0 (no)
109	00	desigidx	0
		localtimetype[1]	
110	00 00 00 00	utoff	0 (+00:00)
114	00	isdst	0 (no)
115	04	desigidx	4
116	2d 30 30 00	designations[0]	"-00\0"
120	47 4d 54 00	designations[4]	"GMT\0"
		leapsecond[0]	

File Offset	Hexadecimal Octets	Record Name / Field Name	Field Value
124	00 00 00 00 58 68 46 9a	occurrence	1483228826 (2016-12-31T23:59:60Z)
132	00 00 00 1b	correction	27
		leapsecond[1]	
136	00 00 00 00 66 7d fd 1b	occurrence	1719532827 (2024-06-28T00:00:01Z)
144	00 00 00 1b	correction	27
148	0a	NL	'\n'
149	47 4d 54 30 42 53 54 2c 4d 33 2e 35 2e 30 2f 31 2c 4d 31 30 2e 35 2e 30	TZ string	"GMT0BST,M3.5.0/1,M10.5.0"
173	0a	NL	'\n'

Table 5

Appendix C. Changes from RFC 8536

- Added definition of Leap Second.
- Added specification of the version 4 format and the optional leap-second table truncation and expiration, along with an example and relevant interoperability considerations.
- Documented the longstanding practice that UT with designation string "-00" denotes unspecified local time. Added recommendation that this designation string should be used for timestamps excluded by TZif file truncation.
- Required support in version 2 files for all-year daylight saving time, using POSIX TZ strings with negative DST, as this is not an extension to POSIX (Section 3.3.1).
- Applied erratum [Err6435].
- Addressed errata [Err6426] and [Err6757] as well as several other errors in the examples.
- Added additional interoperability considerations and common issues.
- Added an example of a TZif file truncated at the end (Appendix B.3).
- Added informational notes to Appendix B.4.
- Miscellaneous editorial changes.

Acknowledgments

The authors would like to thank the following individuals for contributing their ideas and support for writing this specification: Michael Douglass, Ned Freed, Guy Harris, Eliot Lear, Alexey Melnikov, and Tim Parenti.

Authors' Addresses

Arthur David Olson
Email: arthurdavidolson@gmail.com

Paul Eggert
University of California, Los Angeles
Email: eggert@cs.ucla.edu

Kenneth Murchison
Fastmail US LLC
Email: murch@fastmailteam.com