

RFC 9636 : The Time Zone Information Format (TZif)

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 31 octobre 2024

Date de publication du RFC : Octobre 2024

<https://www.bortzmeyer.org/9636.html>

Ce nouveau RFC documente un format déjà ancien et largement déployé, TZif, un format de description des fuseaux horaires. Il définit également des types MIME pour ce format, `application/tzif` et `application/tzif-leap`. Il remplace le premier RFC de normalisation de ce format, le RFC 8536¹, mais il y a très peu de changements. Bienvenue donc à la version 4 du format, spécifiée dans ce RFC.

Ce format existe depuis quarante ans (et a pas mal évolué pendant ce temps) mais n'avait apparemment jamais fait l'objet d'une normalisation formelle avant le RFC 8536 en 2019. La connaissance des fuseaux horaires est indispensable à toute application qui va manipuler des dates, par exemple un agenda. Un fuseau horaire se définit par un décalage par rapport à UTC, les informations sur l'heure d'été, des abréviations pour désigner ce fuseau (comme CET pour l'heure de l'Europe dite « centrale ») et peut-être également des informations sur les secondes intercalaires. Le format iCalendar du RFC 5545 est un exemple de format décrivant les fuseaux horaires. TZif, qui fait l'objet de ce RFC, en est un autre. Contrairement à iCalendar, c'est un format binaire.

TZif vient à l'origine du monde Unix et est apparu dans les années 1980, quand le développement de l'Internet, qui connecte des machines situées dans des fuseaux horaires différents, a nécessité que les machines aient une meilleure compréhension de la date et de l'heure. Un exemple de source faisant autorité sur les fuseaux horaires est la base de l'IANA <<https://www.iana.org/time-zones>> décrite dans le RFC 6557 et dont l'usage est documenté à l'IANA <<https://www.iana.org/time-zones/repository/tz-link.html>>. Pour la récupérer, voir par exemple le RFC 7808.

La section 2 de notre RFC décrit la terminologie du domaine :

- Temps Universel Coordonné (UTC est le sigle officiel) : la base du temps légal. Par exemple, en hiver, la France métropolitaine est en UTC + 1. (GMT n'est utilisé que par les nostalgiques de l'Empire britannique.)

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc8536.txt>

- Heure d'été (le terme français est incorrect, l'anglais DST - "*Daylight Saving Time*" est plus exact) : le décalage ajouté ou retiré à certaines périodes, pour que les activités humaines, et donc la consommation d'énergie, se fassent à des moments plus appropriés (cette idée est responsable d'une grande partie de la complexité des fuseaux horaires).
- Temps Atomique International (TAI) : contrairement à UTC, qui suit à peu près le soleil, TAI est déconnecté des phénomènes astronomiques. Cela lui donne des propriétés intéressantes, comme la prédictibilité (alors qu'on ne peut pas savoir à l'avance quelle sera l'heure UTC dans un milliard de secondes) et la monotonie (jamais de sauts, jamais de retour en arrière, ce qui peut arriver à UTC). Cela en fait un bon mécanisme pour les ordinateurs <<https://www.bortzmeyer.org/leap-second-or-not-leap-second.html>>, mais moins bon pour les humains qui veulent organiser un pique-nique. Actuellement, il y a 37 secondes de décalage entre TAI et UTC.
- Secondes intercalaires : secondes ajoutées de temps en temps à UTC pour compenser les variations de la rotation de la Terre.
- Correction des secondes intercalaires : TAI - UTC - 10 (lisez le RFC pour savoir pourquoi 10). Actuellement 27 secondes.
- Heure locale : l'heure légale en un endroit donné. La différence avec UTC peut varier selon la période de l'année, en raison de l'heure d'été. En anglais, on dit aussi souvent « le temps au mur » ("*wall time*") par référence à une horloge accrochée au mur. Quand on demande l'heure à M. Toutlemonde, il donne cette heure locale, jamais UTC ou TAI ou le temps Unix.
- "*Epoch*" : le point à partir duquel on compte le temps. Pour Posix, c'est le 1 janvier 1970 à 00h00 UTC.
- Temps standard : la date et heure « de base » d'un fuseau horaire, sans tenir compte de l'heure d'été. En France métropolitaine, c'est UTC+1.
- Base de données sur les fuseaux horaires : l'ensemble des informations sur les fuseaux horaires (cf. par exemple RFC 7808). Le format décrit dans ce RFC est un des formats possibles pour une telle base de données.
- Temps universel : depuis 1960, c'est équivalent à UTC, mais le RFC préfère utiliser UT.
- Temps Unix : c'est ce qui est renvoyé par la fonction `time()`, à savoir le nombre de secondes depuis l'"*epoch*", donc depuis le 1 janvier 1970. Il ne tient pas compte des secondes intercalaires, donc il existe aussi un « temps Unix avec secondes intercalaires » (avertissement : tout ce qui touche au temps et aux calendriers est compliqué <<https://www.bortzmeyer.org/calendrical-calculations.html>>). C'est ce dernier qui est utilisé dans le format TZif, pour indiquer les dates et heures des moments où se fait une transition entre heure d'hiver et heure d'été.

La section 3 de notre RFC décrit le format lui-même. Un fichier TZif est composé d'un en-tête (taille fixe de 44 octets) indiquant entre autres le numéro de version de TZif. La version actuelle est 4. Ensuite, on trouve les données. Dans la version 1 de TZif, le bloc de données indiquait les dates de début et de fin des passages à l'heure d'été sur 32 bits, ce qui les limitait aux dates situées entre 1901 et 2038. Les versions ultérieures de TZif sont passées à 64 bits, ce qui permet de tenir environ 292 milliards d'années mais le bloc de données de la version 1 reste présent, au cas où il traîne encore des logiciels ne comprenant que la version 1. Notez que ces 64 bits permettent de représenter des dates antérieures au Big Bang, mais certains logiciels ont du mal avec des valeurs situées trop loin dans le passé.

Les versions 2, 3 et 4 ont un second en-tête de 44 octets, et un bloc de données à elles. Les vieux logiciels arrêtent la lecture après le premier bloc de données et ne sont donc normalement pas gênés par cet en-tête et ce bloc supplémentaires. Les logiciels récents peuvent sauter le bloc de données de la version 1, qui ne les intéresse a priori pas (voir section 4 et annexe A). C'est au créateur du fichier de vérifier que les blocs de données destinés aux différentes versions sont raisonnablement synchrones, en tout cas pour les dates antérieures à 2038.

Nouveauté apparue avec la version 2, il y a aussi un pied de page à la fin. Les entiers sont stockés en gros boutien, et en complément à deux. L'en-tête commence par la chaîne magique « TZif » (U+0054 U+005A U+0069 U+0066), et comprend la longueur du bloc de données (qui dépend du nombre de transitions, de secondes intercalaires et d'autres informations à indiquer). Le bloc de données contient la liste

des transitions, le décalage avec UT, le nom du fuseau horaire, la liste des secondes intercalaires, etc. Vu par le mode hexadécimal <https://www.bortzmeyer.org/emacs-mode-hexa.html> d'Emacs, voici le début d'un fichier Tzif version 2 (pris sur une Ubuntu, dans /usr/share/zoneinfo/Europe/Paris). On voit bien la chaîne magique, puis le numéro de version, et le début du bloc de données :

```
00000000: 545a 6966 3200 0000 0000 0000 0000 0000 TZif2.....
00000010: 0000 0000 0000 000d 0000 000d 0000 0000 .....
00000020: 0000 00b8 0000 000d 0000 001f 8000 0000 .....
00000030: 9160 508b 9b47 78f0 9bd7 2c70 9cbc 9170 .`P..Gx...,p...p
00000040: 9dc0 48f0 9e89 fe70 9fa0 2af0 a060 a5f0 ..H....p...`..
...
```

Avec `od`, ça donnerait :

```
% od -x -a /usr/share/zoneinfo/Europe/Paris
00000000  5a54  6669  0032  0000  0000  0000  0000  0000
          T   Z   i   f   2 nul nul nul nul nul nul nul nul nul
00000020  0000  0000  0000  0000  0d00  0000  0d00  0000  0000
          nul nul nul nul nul nul nul cr nul nul nul cr nul nul nul
00000040  0000  b800  0000  0d00  0000  1f00  0080  0000
          nul nul nul 8 nul nul nul cr nul nul nul us nul nul nul
00000060  6091  8b50  479b  f078  d79b  702c  bc9c  7091
          dc1 ` P vt esc G x p esc W , p fs < dc1 p
...
```

Des exemples détaillés et commentés de fichiers TZif figurent en annexe B. À lire si vous voulez vraiment comprendre les détails du format.

Le pied de page indique notamment les extensions à la variable d'environnement TZ https://www.gnu.org/software/libc/manual/html_node/TZ-Variable.html. Toujours avec le mode hexadécimal d'Emacs, ça donne :

```
00000b80: 4345 542d 3143 4553 542c 4d33 2e35 2e30 CET-1CEST,M3.5.0
00000b90: 2c4d 3130 2e35 2e30 2f33 0a          ,M10.5.0/3.
```

On a vu que le format TZif avait une histoire longue et compliquée. La section 4 du RFC est entièrement consacrée aux problèmes d'interopérabilité, liés à la coexistence de plusieurs versions du format, et de beaucoup de logiciels différents. Le RFC conseille (sections 4 et 5) :

- De ne plus générer de fichiers suivant la version 1, qui ne marchera de toute façon plus après 2038.
- Les logiciels qui en sont restés à la version 1 doivent faire attention à arrêter leur lecture après le premier bloc (dont la longueur figure dans l'en-tête).
- La version 4 n'apporte pas beaucoup par rapport à la 2 et à la 3 et donc, sauf si on utilise les nouveautés spécifiques de la 4, il est recommandé de produire plutôt des fichiers conformes à la version 2 ou 3.
- Un fichier TZif transmis via l'Internet devrait être étiqueté `application/tzif-leap` ou `application/tzif` (s'il n'indique pas les secondes intercalaires). Ces types MIME sont désormais dans le registre officiel <https://www.iana.org/assignments/media-types/media-types.xml#application> (cf. section 9 du RFC).

L'annexe A du RFC en rajoute, tenant compte de l'expérience accumulée ; par exemple, certains lecteurs de TZif n'acceptent pas les noms de fuseaux horaires contenant des caractères non-ASCII et il peut donc être prudent de ne pas utiliser ces caractères. Plus gênant, il existe des lecteurs assez bêtes pour planter lorsque des temps sont négatifs. Or, les entiers utilisés dans TZif sont signés, afin de pouvoir représenter les moments antérieurs à l'"epoch". Donc, attention si vous avez besoin de données avant le premier janvier 1970, cela perturbera certains logiciels bogués.

En parlant des noms de fuseaux horaires, le RFC rappelle (section 5) que le sigle utilisé dans le fichier Tzif (comme « CET ») est en anglais et que, si on veut le traduire, cela doit être fait dans l'application, pas dans le fichier. Le RFC donne comme exemple CST, qui peut être présenté comme « HNC », pour « Heure Normale du Centre ».

Autre piège avec ces sigles utilisés pour nommer les fuseaux horaires : ils sont ambigus. PST peut être la côte Ouest des USA mais aussi l'heure du Pakistan ou des Philippines.

La section 7 du RFC donne quelques conseils de sécurité :

- L'en-tête indique la taille des données mais le programme qui lit le fichier doit vérifier que ces indications sont correctes, et n'envoient pas au-delà de la fin du fichier.
- TZif, en lui-même, n'a pas de mécanisme de protection de l'intégrité, encore moins de mécanisme de signature. Il faut fournir ces services extérieurement (par exemple avec curl, en récupérant via HTTPS).

Enfin, l'annexe C du RFC liste les changements depuis le RFC 8536, changements qui mènent à cette nouvelle version, la 4. Rien de crucial mais :

- Formalisation de l'ancienne convention comme quoi un décalage de « - 0 » avec UTC indique qu'on ne connaît pas le vrai décalage.
- Correction de plusieurs erreurs <<https://www.rfc-editor.org/errata/rfc8536>>.

Une bonne liste de logiciels traitant ce format figure à l'IANA <<https://data.iana.org/time-zones/tz-link.html>>.