

RFC 9639 : Free Lossless Audio Codec

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 19 décembre 2024

Date de publication du RFC : Décembre 2024

<https://www.bortzmeyer.org/9639.html>

Le format FLAC, normalisé dans ce RFC, permet de stocker et de transmettre du son sans perte (contrairement à des mécanismes de compression comme MP3). Il est donc utile, par exemple pour l'archivage à long terme (mais aussi pour la haute fidélité).

Le multimédia, c'est compliqué. Le RFC fait près de 100 pages, pour un format audio qui se veut pourtant simple. FLAC fait de la compression pour limiter l'espace de stockage nécessaire mais en prenant soin de permettre une décompression qui redonne exactement le fichier originel (d'où le « sans perte » dans son nom, qu'il partage avec d'autres formats, comme le FFV1 du RFC 9043¹). Je ne connais pas assez le domaine de l'audio pour faire un résumé intelligent du RFC donc je vous invite à le lire si vous voulez comprendre comment FLAC fonctionne (les sections 4 à 6 vous font une description de haut niveau).

Si on prend un fichier FLAC, un programme comme `file` peut le comprendre :

```
% file Thélème.flac
Thélème.flac: FLAC audio bitstream data, 24 bit, stereo, 44.1 kHz, length unknown
```

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc9043.txt>

Le fait que l'échantillonnage soit à 44,1 kHz est encodé dans les métadonnées (cf. section 9.1.2 du RFC), que `file` a pu lire (section 9.1.3 pour le fait que le fichier soit en stéréo).

Il y a quelques fichiers FLAC sur Wikimedia Commons <https://commons.wikimedia.org/wiki/Category:FLAC_files>. Mais un bon exemple de l'utilisation de FLAC est donné par l'enregistrement des Variations Goldberg par Kimiko Ishizaka, que vous pouvez télécharger en FLAC <<https://archive.org/details/OpenGoldbergVariations>> et sous une licence libre (CC0).

La mise en œuvre de référence de FLAC est en logiciel libre et se nomme libFLAC <<https://xiph.org/flac/>>. Mais il existe beaucoup d'autres programmes qui savent gérer FLAC (j'ai écouté plusieurs fichiers FLAC avec `vlc` pour cet article), le lire, l'écrire, l'analyser, etc. Une liste est disponible <<https://github.com/ietf-wg-cellar/flac-specification/wiki/Implementations>> sur le site du groupe de travail IETF. Il faut dire que FLAC, bien qu'il vienne seulement d'être normalisé, est un format ancien, dont le développement a commencé en 2000, et il n'est donc pas étonnant que les programmeur-ses aient eu le temps de travailler.

Si vous voulez ajouter du code à cette liste, lisez bien la section 11 du RFC, sur la sécurité. Un programme qui lit du FLAC doit être paranoïaque (comme avec n'importe quel autre format, bien sûr) et se méfier des cas pathologiques. Ainsi, le RFC note qu'on peut facilement créer un fichier FLAC de 49 octets qui, décomprimé, ferait 2 mégaoctets, ce qui pourrait dépasser la mémoire qui avait été réservée. Ce genre de surprises peut arriver à plusieurs endroits. L'annexe D du RFC contient plusieurs fichiers FLAC intéressants (et vous pouvez les retrouver en ligne <<https://github.com/ietf-wg-cellar/flac-specification>>) et il existe également une collection de fichiers FLAC <<https://github.com/ietf-wg-cellar/flac-test-files>>, dont certains sont délibérément anormaux, pour que vous puissiez tester la robustesse de votre programme. Autre piège (mais lisez toute la section 11, il y en a beaucoup), FLAC permet de mettre des URL dans les fichiers, URL qu'il ne faut évidemment pas déréférencer bêtement.

Ce RFC est également la documentation pour l'enregistrement <<https://www.iana.org/assignments/media-types/media-types.xml#audio>> du type MIME `audio/flac` (section 12.1).

Pour une analyse technique de la prétention « sans perte » de FLAC, voir cet article de Guillaume Seznec <<https://aerogus.net/posts/flac-format-audio-vraiment-sans-perte/>>.