ChatGPT

Format ZIP et variantes

Le format ZIP original (développé par Phil Katz en 1989) est limité par des entiers 32 bits : il ne peut gérer qu'au maximum 2^32–1 octets (~4 Go) par fichier compressé, 2^32–1 octets au total et 65 535 entrées dans l'archive [1]. La norme ZIP 4.5 (2009) introduit l'extension **ZIP64** pour lever ces limites : les tailles passent à 2^64–1 (16 EiB) et le nombre de fichiers à 2^64–1 [1], via des champs supplémentaires (extra-fields) dans l'en-tête. Par ailleurs, WinZip a introduit en 2008 l'extension **".ZIPX"** : ce nouveau format (géré notamment par WinZip 12.1+) permet d'utiliser dans l'archive d'autres algorithmes de compression (BZip2, LZMA, PPMd…) ou de chiffrement, tout en conservant la compatibilité avec les logiciels non ZIPX en changeant l'extension de fichier [2] [3]. D'autres variantes existent, comme « Deflate64 » (méthode alternative avec plus grande fenêtre), les archives multi-volumes (spanning) ou auto-extractibles (SFX), mais le ZIP standard et ses extensions ZIP64/ZIPX restent les plus répandues.

- **ZIP standard (v2.0)** – Limité à 4 GiB/fichier et 65 535 fichiers, compression Deflate (LZ77+Huffman) par défaut [1].
- **ZIP64 (v4.5)** – Extension de la norme pour fichiers très volumineux (tailles sur 64 bits). Remplace les champs de taille 32 bits par 0xFFFFFFFF et stocke les vraies valeurs dans des extra-fields [4] [1]. Permet jusqu'à 2^64–1 octets et 2^64–1 fichiers.
- **ZIPX (WinZip)** – Format propre à WinZip avec extension .zipx. Autorise l'usage de nouvelles méthodes (BZip2, LZMA, PPMd, JPEG, WavPack) pour améliorer le ratio, au prix d'une compatibilité réduite avec les outils ZIP classiques [2] [3].
- **Autres variantes** – Certaines implémentations (p.ex. Info-ZIP) ont ajouté des options (UTF-8 pour noms, chiffrement AES-128/256, etc.), mais ce sont plutôt des extensions logicielles qu'un nouveau format.

Implémentations logicielles de ZIP

Plusieurs logiciels prennent en charge le format ZIP, avec des profils fonctionnels différents :

- **WinZip** – Archiver commercial historique sur Windows. Supporte parfaitement ZIP standard et ZIP64, avec chiffrement AES et intégration de méthodes avancées via le format ZIPX [2] [5]. Performances correctes en Deflate, mais inférieures à 7-Zip : selon Tom's Hardware, 7-Zip dépasse WinZip tant en ratio de compression qu'en vitesse [6]. WinZip offre une interface riche (compression multithread, assistant cloud, etc.), mais c'est une solution payante et Windows-centric.
- **7-Zip** – Open source (LGPL), disponible sous Windows et Linux (via p7zip). Propose son propre format 7z très performant (LZMA2) mais gère aussi ZIP (lecture/écriture). Peut créer des .zip en Deflate, BZip2, LZMA, PPMd, Zstandard (depuis v19.00) [5]. En pratique, 7-Zip atteint souvent de meilleurs ratios que WinZip sur ZIP Deflate (bien que le résultat soit proche) et compresse/décompresse plus vite [6]. 7-Zip est très populaire pour son rapport qualité-vitesse.
- **Info-ZIP (zip/unzip)** – Logiciels en ligne de commande open source portables (BSD-like) très utilisés sous Unix/Linux. Zip 3.0 (2008) a ajouté la prise en charge de ZIP64, d'archives multi-volumes, de la compression BZip2 et des noms Unicode [7]. Seules les méthodes Deflate et BZip2 sont disponibles nativement ; le support de LZMA/PPMd n'est apparu que dans des versions bêta récentes (zip 3.1c/6.10b) [5]. Info-ZIP est réputé pour sa compatibilité et sa robustesse, mais peut être plus lent que 7-Zip. Il ne supporte pas (sans patch) les algorithmes récents comme Zstandard.

- **Autres archivers** – De nombreux outils multi-format (PeaZip, Bandizip, PowerArchiver, etc.) gèrent ZIP, certains utilisent des bibliothèques internes (7-Zip, zlib, libarchive…). Les utilitaires intégrés (Windows Explorer, macOS Archive Utility, Java/.NET libs) lisent le plus souvent le ZIP standard et ZIP64, mais peuvent ne pas reconnaître les méthodes avancées (PPMd, Zstd).

Performances comparées : 7-Zip est régulièrement cité comme « champion » en vitesse et ratio [6]. WinZip est rapide mais un peu moins performant, et Info-ZIP est robuste et portable. Par exemple, en Deflate maximum, 7-Zip compresse légèrement moins que WinZip (97,7 Mo vs 97,1 Mo sur un jeu de test) [8], mais 7-Zip compresse bien plus vite. L'algorithme BZip2 (pris en charge par WinZip, 7-Zip, Info-ZIP) offre un meilleur ratio (≈80 Mo dans le test) au prix d'une compression plus lente [8]. En résumé, **7-Zip** tend à fournir les meilleures performances globales [6], **WinZip** apporte une interface riche et le format ZIPX propriétaire, et **Info-ZIP** assure compatibilité maximale (notamment sur plates-formes Unix).

Algorithmes de compression

Le format ZIP peut encapsuler plusieurs algorithmes (méthodes) de compression :

- **Deflate (méthode 8)** – Algorithme standard du ZIP (créé pour PKZIP), combinaison de **LZ77** (fenêtre glissante) et de codage de Huffman [9]. Il est rapide en compression/décompression et universel (utilisé aussi en gzip, PNG, HTTP, etc.), mais son ratio est modéré (suffisant pour la plupart des usages).
- **BZip2 (méthode 12)** – Utilise la *transformation de Burrows-Wheeler* sur des blocs (jusqu'à 900 kB), suivie d'une *codage par Huffman* sophistiqué. Offre un ratio bien supérieur à Deflate sur des données textuelles ou redondantes [10], mais la compression est lente et gourmande en mémoire. Idéal quand on peut compresser hors ligne (backups, distributions), moins adapté pour traitement temps réel.
- **LZMA (méthode 14)** – « Lempel–Ziv–Markov chain Algorithm », utilisé par 7-Zip. C'est un algorithme **LZ77** très évolué, avec dictionnaire de très grande taille et codage *range* (entropy coding). Il donne un très fort taux de compression (meilleur que Deflate/BZip2) sur de gros fichiers, au prix d'un coût CPU/mémoire élevé (surtout à la compression). La décompression reste rapide. LZMA a été introduit dans le ZIP par certaines versions de WinZip et 7-Zip (ZIPX) et dans les dernières mises à jour d'Info-ZIP [5]. Il est à la fois performant (ratio) et « solide » (résistance aux corruptions) [11].
- **PPMd (méthode 98)** – *Prediction by Partial Matching*, une compression contextuelle statistique (variant de PPM). PPMd, version de Dmitry Shkarin, modélise les contextes récents pour prédire chaque octet suivant. Il excelle sur les données textuelles (taux de compression très élevé), mais exige beaucoup de mémoire et de temps de calcul. Souvent utilisé en archivation haute compression (ex. RAR, ZIPX de WinZip). Présent dans ZIP comme option (WinZip ZIPX) et a été ajouté dans Info-ZIP en 2015 pour la création de ZIPX [5].
- **Zstandard (Zstd, méthode 20/93)** – Algorithme moderne de Facebook (publié en RFC 8878). Conçu pour être *très rapide* tout en fournissant de bons ratios (mieux que zlib/Deflate à vitesse équivalente) [12]. Zstd a un décodeur extrêmement rapide (plusieurs GB/s), un compresseur réglable du très rapide au très compact, et gère des dictionnaires pour très petits fichiers. Devenu disponible dans le ZIP depuis la spécification 6.3.7 (2019) [13], puis déplacé à l'ID 93 en 2020. Zstd est idéal quand on veut le compromis maximal vitesse/ratio.

Chaque algorithme a ses cas d'usage : Deflate pour la compatibilité universelle, BZip2/LZMA/PPMd pour la compression maximale (archives historiques, sources), et Zstd pour les scénarios temps réel et gros volumes. Notons aussi la méthode **Deflate64** (id 9) – variante de Deflate avec fenêtre 64 kB – peu utilisée aujourd'hui.

Formats conteneurs basés sur ZIP

De nombreux formats de fichiers populaires sont en réalité des archives ZIP structurées : ils définissent une arborescence particulière à l'intérieur du ZIP pour stocker des données. Par exemple :

- **DOCX, XLSX, PPTX (Office Open XML)** – Documents Microsoft Office 2007+. Ce sont des archives ZIP (« Open Packaging Convention ») qui contiennent un fichier `[Content_Types].xml` à la racine et des dossiers `_rels/` (relations) et `docProps/`, plus un dossier spécifique (`word/` pour DOCX, `xl/` pour XLSX, `ppt/` pour PPTX). Par exemple, `word/document.xml` dans DOCX stocke le contenu texte principal [14].
- **ODT, ODS, ODP (OpenDocument)** – Documents OpenOffice/LibreOffice. Archivent plusieurs fichiers XML : `mimetype` (texte spécial « application/vnd.oasis.opendocument.text » pour ODT) au premier niveau [15], puis dans `META-INF/manifest.xml` la liste des parties (par ex. `content.xml` pour le contenu, `styles.xml`, `settings.xml`, etc.) [15] [16].
- **EPUB (eBooks)** – Standard pour livres numériques. Archive ZIP qui doit contenir en entête un fichier `mimetype` valant `application/epub+zip` [17]. Puis le dossier `META-INF/container.xml` pointe vers le `package.opf` (dans `OEBPS/`) qui décrit la structure (manifest des fichiers XHTML, spine pour l'ordre de lecture) [18].
- **JAR/ WAR (Java Archive)** – Format Java. C'est un ZIP contenant au minimum un répertoire `META-INF/` avec le fichier `MANIFEST.MF` (manifest) et des fichiers .class ou ressources. Les manifest de JAR peuvent contenir des métadonnées (classe principale, etc.) [19].
- **APK (Android Package)** – Archivage des applications Android. Un APK est un ZIP comprenant notamment : `AndroidManifest.xml` (binaire XML décrivant l'application), `classes.dex` (bytecode Dalvik), `resources.arsc` (ressources compilées), les dossiers `lib/ABI/` (librairies natives), `res/` (images/layouts), `assets/` (ressources brutes), et `META-INF/` (certificat et signature de l'app) [20] [21].
- **Autres** – D'autres formats (XPI pour extensions Firefox, IPA pour iOS, OpenXML `.rels` / `.bin`, etc.) reposent eux aussi sur ZIP. Chacun suit ses conventions (noms de fichiers clés, structures imbriquées) pour que le logiciel correspondant puisse lire son contenu.

Structure interne d'un fichier ZIP

Un fichier ZIP est organisé en *en-têtes locaux* (pour chaque fichier), suivi des données compressées, et se termine par le *central directory* et un enregistrement de fin. Chaque entrée de l'archive commence par un **en-tête local** (Local File Header, signature `0x04034b50`) contenant le nom du fichier, l'algorithme de compression, la date de modification, la somme de contrôle CRC-32 et les tailles non compressée et compressée (32 bits) [4]. Les données compressées du fichier suivent immédiatement cet en-tête (le CRC et tailles peuvent être répétées dans un *data descriptor* optionnel en fin de données si le flag correspondant est activé [22]).

Après toutes les entrées, vient le **Central Directory**, un registre général référençant chaque fichier. Chaque enregistrement du central directory (signature `0x02014b50`) récapitule le nom de fichier, la méthode, le CRC-32, les tailles et – surtout – l'offset vers son en-tête local correspondant [4] [23]. Le central directory permet d'énumérer rapidement tous les fichiers sans remonter tout l'archive. Enfin, le fichier se termine par l'enregistrement **End Of Central Directory** (EOCD, signature `0x06054b50` [24]) qui indique notamment le nombre total de fichiers, la taille et l'offset du central directory, et la longueur d'un éventuel commentaire. Ce record est toujours positionné à la fin du ZIP (on le trouve en balayant l'archive depuis la fin).

Tous les fichiers inclus sont protégés par une vérification CRC-32 (polynôme standard IEEE 802.3 [4]) : chaque en-tête local et central stocke la CRC-32 sur les données. Par défaut, les tailles (CRC, tailles compressée/non) sont sur 32 bits. Si une valeur dépasse 0xFFFFFFFF (cas ZIP64), le champ 32 bits vaut 0xFFFFFFFF et la vraie valeur 64 bits est inscrite dans les *extra-fields* de l'en-tête (ZIP64) [4] [1]. Ainsi, ZIP64 étend la compatibilité tout en restant lisible par les anciens décompresseurs (ignorant les extra-fields). En résumé, la structure se lit du début (en-têtes locaux) vers la fin (central directory + EOCD), avec des signatures fixes pour chaque bloc [25] [24].

Sources : Spécifications officielles PKWARE (APPNOTE), Wikipedia ZIP, et analyses techniques [25] [4] [24] [1]. Les détails pratiques proviennent de tests et benchmarks open-source [8] [6].

---

[1] [3] [13] ZIP (file format) - Wikipedia
https://en.wikipedia.org/wiki/ZIP_(file_format)

[2] ZIP file format
https://peazip.github.io/zip-file-format.html

[4] [22] [23] The structure of a PKZip file
https://users.cs.jmu.edu/buchhofp/forensics/formats/pkzip.html

[5] [7] Info-ZIP - Wikipedia
https://en.wikipedia.org/wiki/Info-ZIP

[6] And The Undisputed Winner Is... - Compression Performance: 7-Zip, MagicRAR, WinRAR, WinZip | Tom's Hardware
https://www.tomshardware.com/reviews/winrar-winzip-7-zip-magicrar,3436-13.html

[8] Compression benchmark: 7-Zip, PeaZip, WinRar, WinZip comparison
https://peazip.github.io/peazip-compression-benchmark.html

[9] Deflate - Wikipedia
https://en.wikipedia.org/wiki/Deflate

[10] Bzip2: High-Quality Compression with Burrows-Wheeler Algorithm | Lenovo CA
https://www.lenovo.com/ca/en/glossary/bzip2/?srsltid=AfmBOooZQhQUVmI7IUBtmsEuT4_tBX8FvPzSmjWUOnNCuJUwobd1Qvzd

[11] LZMA - Lempel Ziv Markov Chain Algorithm
https://products.aspose.com/zip/most-common-archives/what-is-lzma/

[12] Zstandard - Real-time data compression algorithm
http://facebook.github.io/zstd/

[14] Office Open XML file formats - Wikipedia
https://en.wikipedia.org/wiki/Office_Open_XML_file_formats

[15] [16] How ODT files are structured | Opensource.com
https://opensource.com/article/22/8/odt-files

[17] [18] A look inside an EPUB file | Opensource.com
https://opensource.com/article/22/8/epub-file

[19] JAR File Specification
https://docs.oracle.com/javase/8/docs/technotes/guides/jar/jar.html

[20] [21] apk (file format) - Wikipedia
https://en.wikipedia.org/wiki/Apk_(file_format)

24 25 The ZIP File Format. This is a technical overview of the... | by Felix Stridsberg | Medium
https://medium.com/@felixstridsberg/the-zip-file-format-6c8a160d1c34