

# [Sustainability of Digital Formats: Planning for Library of Congress Collections](#)



[Introduction](#) | [Sustainability Factors](#) | [Content Categories](#) | [Format Descriptions](#) | [Contact](#)  
[Format Description Categories](#) >> [Browse Alphabetical List](#)

## ZIP File Format (PKWARE)

[>> Back](#)

### Table of Contents

- [Identification and description](#)
- [Local use](#)
- [Sustainability factors](#)
- [Quality and functionality factors](#)
- [File type signifiers](#)
- [Notes](#)
- [Format specifications](#)
- [Useful references](#)

### Format Description Properties

- ID: fdd000354
- Short name: ZIP\_PK
- Content categories: aggregate
- Format Category: family
- Other facets: container-bundle, binary, structured, compression
- Last significant FDD update: 2020-05-28
- Draft status: Full

### Identification and description

Full name	ZIP File Format (PKWARE)
Description	<p>This format is designed for cross-platform data exchange and efficient data storage for a set of related files. ZIP_PK is a de facto industry standard, developed, maintained, and openly documented by <a href="#">PKWARE</a>. The original version of the format was developed by Phil Katz (hence the "PK" in PKWARE). ZIP_PK combines data compression, file management, and data encryption within a portable archive format. A ZIP file is a package containing one or more files, usually compressed and sometimes encrypted. The basic structure consists of a sequence of chunks comprising a "local file header" followed by the file data (after compression and/or encryption) followed by a chunk known as the "central directory," which lists the files in the package along with key metadata to support their extraction, decryption, etc. Over the years the format has been extended by PKWARE to adapt to new expectations of users and to changing technology, for example, to allow improved methods for compression and encryption, to support larger file sizes, and to support UNICODE UTF-8 for filenames. See <a href="#">Notes</a> below for more detail on chronological versions.</p> <p>Over the years a few variants of the ZIP format have been introduced by other companies, variants that were not necessarily fully compatible with any particular ZIP_PK version, with the differences often being related to timing of introduction of new features, such as support for long file names or stronger encryption. For clarity, the compilers of this resource have chosen to use the name "ZIP_PK" for the format described in this document. See <a href="#">Notes</a> below for a brief discussion of variants.</p>

	<p>ZIP_PK has been adopted as the package or container for several digital formats that represent a single document or other logical unit but comprise multiple files. These include:</p> <ul style="list-style-type: none"> <li>• Office Open XML (<a href="#">OOXML</a>, standardized as ECMA 376 and ISO/IEC 29500)</li> <li>• Office Document Format (<a href="#">ODF</a>, from OASIS, later standardized as ISO/IEC 26300)</li> <li>• EPUB (for electronic publications, originally developed by the International Digital Publishing Forum (IDPF) and since 2017 under the auspices of the World Wide Web Consortium (W3C). See <a href="#">EPUB_3_0</a> (ISO/IEC TS 30135:2014) and <a href="#">EPUB_3_2</a>.)</li> <li>• Java ARchive (JAR, used to distribute software applications or libraries in the Java programming language).</li> </ul>
<b>Production phase</b>	May be used at any lifecycle phase for bundling/packaging files together for exchange, storage, or distribution. In particular, XML-based office productivity formats use a ZIP package that may be used during the initial creating phase; and XML-based formats such as EPUB, intended for distributing published (final-stage) documents, also use ZIP as a package or container.
<b>Relationship to other formats</b>	
<b>Has subtype</b>	<a href="#">ZIP_6_2_0</a> , ZIP file format, version 6.2.0 (PKWARE). Version 6.2.0 of the APPNOTE.TXT from PKWARE, published in 2004, has been incorporated into Office OpenXML (OOXML, ISO 29500-2:2008-2012) and the Open Document Format (ODF: 1.0 and 1.2 as published by OASIS) as the document container format.
<b>Has subtype</b>	<a href="#">ZIP_6_3_3</a> , ZIP File Format, Version 6.3.3 (PKWARE). Chosen as the basis for a proposed ISO specification (to be ISO/IEC 21320, once approved) for a Document Container Format.
<b>Has subtype</b>	<a href="#">ZIP_21320_1</a> , Document Container File: Core (based on ZIP 6.3.3). ISO/IEC 21320-1 Information technology -- Document Container File -- Part 1: Core
<b>Has subtype</b>	The ZIP_PK format has other chronological versions, more of which may be separately described in the future, for example, if specific versions are incorporated into other specifications.
<b>Has subtype</b>	<a href="#">SLPK</a> , SLPK Scene Layer Package (for I3S layers)
<b>Has subtype</b>	<a href="#">KMZ</a> , KML (formerly Keyhole Markup Language) Zipped
<b>Has subtype</b>	<a href="#">USDZ</a> , Universal Scene Description Zipped (USDZ)
<b>Has subtype</b>	<a href="#">WACZ</a> , Web Archive Collection Zipped
<b>Has subtype</b>	<a href="#">XAP</a> , Silverlight Application Package
<b>Has extension</b>	<a href="#">APK</a> , Android Package
<b>Has extension</b>	<a href="#">IPA</a> , iOS App Store Package
<b>Used by</b>	<a href="#">3MF</a> , 3D Manufacturing Format (3MF)
<b>Used by</b>	<a href="#">IMF_Package</a> , Interoperable Master Format (IMF)
<b>Has modified version</b>	Other entities have introduced variant format versions using the .zip extension, not strictly compatible with any particular chronological version of ZIP_PK, but using its extension capabilities. See <a href="#">Notes</a> below for a brief discussion of variants and compatibility.

## Local use

<b>LC experience or existing holdings</b>	The Library of Congress has several workflows that rely on the ZIP format for transmitting digital content to the library. These workflows rely on open-source toolkits and/or desktop decompressors. In all current workflows, the ZIP format is used for transmission only, with the ZIP wrapper discarded after successful extraction of the contents.
---	---

<b>LC preference</b>	To date, the Library of Congress has found no need to stipulate particular versions of ZIP_PK or particular features of the format to avoid. In general the Library of Congress does not accept content submitted in encrypted form. The Library of Congress Recommended Formats Statement (RFS) lists ZIP as both a preferred (for direct file submission) and acceptable (on a mass storage device) format for packaged delivery of <a href="#">Software and Video Games</a> .
----------------------	--

## Sustainability factors

<b>Disclosure</b>	The ZIP_PK format has been developed and maintained by PKWARE, Inc. since 1998. The format specification is proprietary, but the most recent version has always been openly disclosed as the .ZIP Application Note with a file name of APPNOTE.TXT.
<b>Documentation</b>	<p>In the past, PKWARE provided public access only to the <a href="#">most recent Application Note</a>. In association with the standardization of Office Open XML (OOXML) as ISO 29500:2008, access was also provided to the <a href="#">version 6.2.0</a> explicitly incorporated into that format specification. PKWARE now provides access to <a href="#">an archive of selected versions of the ZIP Application Note</a>.</p> <p>See <a href="#">Notes</a> below for a chronology of versions of APPNOTE.TXT introducing significant new features to the format. Changes between <a href="#">version 6.3.2 (2007)</a> and <a href="#">version 6.3.3 (2012)</a> of APPNOTE.TXT do not reflect changes in the format, but rather editorial and layout changes based on discussions with an ISO working group that studied the ZIP format. Changes, such as adding paragraph numbers, were made to making it easier for ISO standards to reference the ZIP specification in an unambiguous and reliable way.</p>
<b>Adoption</b>	<p>The ZIP_PK format has been very widely adopted. Most usage has been and continues to be based on a subset of features, resulting in a high level of interoperability. Many tools exist to create and read ZIP files that do not include strong encryption, including tools integrated into operating systems for personal computers and/or stand-alone tools shipped with new computers. See <a href="https://en.wikipedia.org/wiki/ZIP_(file_format)">https://en.wikipedia.org/wiki/ZIP_(file_format)</a> for many examples; just a few will be mentioned here.</p> <p>The originator of the format, PKWARE, provides a range of products, from a free <a href="#">ZIP Reader</a> for Windows, <a href="#">PKZIP and SecureZIP</a> for Windows Desktop and various IBM platforms, to server-based products.</p> <p><b>As of September 2012:</b></p> <p>Windows ("compressed folders") and Mac OS (Archive Utility) have built in support for a subset of ZIP features. Earlier versions of these operating systems typically shipped with "lite" versions of utilities such as WinZip and Stuffit. Most versions of Unix (including the version that underlies recent Mac OS versions, ship with <i>zip</i> and <i>unzip</i> utilities, made available from <a href="#">Info-ZIP</a> along with open-source software libraries compatible with many operating systems. The Info-ZIP software, developed starting in 1990 to provide ZIP-compatible tools for operating systems other than DOS, supports the most useful features supported by the ZIP_PK format, and has been widely incorporated into other products, including WinZip and <a href="#">JHOVE (JSTOR/Harvard Object Validation Environment)</a>. Its capabilities have thus served as a common denominator for interoperability. As of 2012, the latest version of Info-ZIP's <i>zip</i> utility is from 2008 and of <i>unzip</i> is from 2009; these versions support filenames in UTF-8 and the ZIP64 extension introduced by PKWARE in 2001 to handle large files. A Java-based utility <i>java.util.zip</i> is included in the Java Developer Kit. Apache Commons Compress extends the functionality of <i>java.util.zip</i>. Several perl and python software libraries for manipulating ZIP files can be found on the web. Note: Many ZIP utilities or software libraries support only a subset of ZIP_PK</p>

	<p>features, particularly those available without licensing. Many compression/archiving utilities, including PKZIP, support ZIP_PK (.zip) as one of several input/output file formats.</p> <p><b>Update on adoption in 2020:</b></p> <p>ZIP support continues to extend. In 2015, <a href="#">ISO/IEC 21320-1</a> was published as an international standard for a container based on the ZIP format for digital documents that comprise multiple files, using a subset of features that can be freely implemented. See <a href="#">ZIP_21320_1</a>. Support for opening ZIP files has been added to iOS. Starting with iOS 7 (2013), ZIP files received via Mail or Messages could be opened. More general support was added in 2018 with the Files app, added in iOS 11. The Files app allows users to see the contents of a ZIP file, and extract files. For Android, a number of apps exist. <a href="#">Comments welcome.</a></p> <p>ZIP_PK has been used as a packaging or container format in other format specifications. These format specifications typically do not incorporate patented features of ZIP_PK and typically constrain the use of other features, for example, allowing only the DEFLATE algorithm for compression. Features that do tend to get incorporated include: long filenames, encoding of filenames, etc. in UTF-8, and the ZIP64 mechanism introduced by PKWARE for handling large ZIP files. See <a href="#">Notes</a> below for more detail.</p>
<b>Licensing and patents</b>	<p>According to <a href="#">Our Founder -- Phil Katz</a>, on the PKWARE website (as of February 2013) the original developer's "decision to dedicate the .ZIP extension and file format specification to the public domain helped the .ZIP file format become a globally open standard." As a result, many utilities for creating or reading files in the ZIP_PK format have been and can be developed without separate license agreements. Two mechanisms supported by more recent versions of the format are covered by PKWARE patents: portions of the strong encryption technology support; and a technique for issuing and incorporating patches to content previously distributed as ZIP archives. Tools that manipulate ZIP files but avoid use of these mechanisms can be developed without license concerns. See <a href="#">Notes</a> below for more detail on the scope of PKWARE patents and licensing.</p> <p>Third parties with rights associated with code incorporated into PKWARE's own ZIP Reader products were listed, with license terms, (as of September 2012) at <a href="http://www.pkware.com/Support-Desktop/zipreaderthirdparty">http://www.pkware.com/Support-Desktop/zipreaderthirdparty</a> (link now via Internet Archive). All were made available to PKZIP under some form of public (open-source) license and would presumably be available to other tool developers. The compilers of this resource are not aware of any patent concerns with compression or encryption methods for which the ZIP_PK specification through version 6.3.3 of APPNOTE.TXT, has identifying codes. <a href="#">Comments welcome.</a></p>
<b>Transparency</b>	Depends upon algorithms and tools to read. Would require sophistication to build tools from scratch.
<b>Self-documentation</b>	<p>Since the format provides a mechanism to bundle files together, a component file that provides rich metadata in a standard or application-specific format can be included in a ZIP file. This practice has been incorporated into the package specifications for EPUB, OOXML, ODF, etc. The ZIP_PK specification includes a central directory, which contains the metadata required to extract the contents of the ZIP file, i.e., file names and directory structure. The inclusion of comments for each file is supported within the format, but not (or only partially) by some utilities or software libraries.</p>
<b>External dependencies</b>	<p>None, beyond the availability of software to extract and decompress the files contained in a ZIP file. The ZIP_PK format itself includes a mechanism for defining within a ZIP file the minimum version that must be supported to extract, decompress, and decrypt (if needed) the contents. Minimum versions for particular extensions and methods for compression and encryption are documented in APPNOTE.TXT. Note: Use of strong</p>

	<p>encryption may rely on keys conveyed externally or external authentication procedures.</p> <p><b>Accessibility Features</b></p> <p>No specific features in the file format. Features to support accessibility would be found in the bundled and compressed files (such as embedded captions and subtitles in audiovisual content, tagged and structured text in textual documents, and alt text for images). Aggregate files can also contain separate files for transcripts, timed text or captions as part of the bundled package. See <a href="#">Relationships to other formats</a> for details.</p>
<b>Technical protection considerations</b>	<p>An important feature of the ZIP_PK format is the ability to encrypt the contained files individually. Over the years, support for increasingly strong forms of encryption, and for encryption of the ZIP central directory has been added. Certain applications that employ the ZIP format to bundle files together, such as OOXML (ISO/IEC 29500-2:2012), explicitly prohibit encryption within the ZIP-based package. Others, such as SCORM and EPUB, employ their own application-specific approaches to encryption.</p>

## Quality and functionality factors

Aggregate	
<b>Compression</b>	Options for compression are extensive and include BZIP2, LZMA, DEFLATE, IBM z/OS CMPSC Compression, IBM TERSE, IBM LZ77 z Architecture, Zstandard (zstd), MP3 Compression, XZ Compression, JPEG variant, WavPack compressed data, PPMd version I, and AE-x encryption marker.
<b>Support for Error Detection</b>	ZIP files may use a CRC-32 hash for checking file integrity.
<b>Beyond normal functionality</b>	Two mechanisms supported by the ZIP_PK format go beyond the functions of bundling and compression. The first is the ability to construct a self-extracting executable .zip file. The second is a "patching" mechanism, described in <a href="#">Notes</a> below.

## File type signifiers and format identifiers

Tag	Value	Note
<b>Filename extension</b>	zip ZIP	Other extensions are used for particular applications that use the ZIP format as a container, for example, <a href="#">EPUB_3</a> , which recommends .epub as the file extension.
<b>Internet Media Type</b>	application/zip	See <a href="#">registration at IANA</a> .
<b>File signature</b>	Hex: 50 4B 03 04 ASCII: P K etx eot	In the specification, this 4-byte "local" file header signature is expressed in little-endian form as 0x04034b50. The compilers of this resource have chosen to express the ZIP signifiers in the byte-order that is consistent with most documentation for internal file signifiers, such as <a href="#">PRONOM</a> and the File Extension Source ( <a href="#">FILExt</a> ). Most ZIP files begin with this 4-byte string, since most ZIP files begin with a local file header. However, as pointed out in <a href="https://en.wikipedia.org/wiki/Zip_(file_format)">https://en.wikipedia.org/wiki/Zip_(file_format)</a> , a ZIP file does not need to begin with a file entry. The signature patterns below should also be considered.
<b>File signature</b>	Hex: 50 4B 01 02 {...} 50 4B 05 06 {...} ASCII: P K soh stx {...} P K enq ack {...}	This signature pattern adapted from PRONOM for convenient display represents the beginning of a central directory record and the mandatory end of central directory record which will be found at or near the end of a ZIP file. See <a href="#">PRONOM record for ZIP (PUID: x-fmt/263)</a> , which documents signatures at both the beginning and end of the file.
<b>File signature</b>	Hex: 50 4B 05 06	At <a href="https://web.archive.org/web/20221112073316/https://www.garykessler.net/library/file_sigs.html">https://web.archive.org/web/20221112073316/https://www.garykessler.net/library/file_sigs.html</a> this signature based on the mandatory <i>end of central directory</i> record is documented as a <i>Trailer</i>



	ASCII: P K enq ack	signature, with a stipulation to expect 18 additional bytes before the end of the file. APPNOTE.TXT 6.3.3 stipulates that a valid file should have one and only one <i>end of central directory</i> record, but does not state explicitly that it be at the physical end of the file.
<b>File signature</b>	Hex: 50 4B 05 06 ASCII: P K enq ack	At beginning of empty ZIP archive file. From <a href="https://web.archive.org/web/20221112073316/https://www.garykessler.net/library/file_sigs.html">https://web.archive.org/web/20221112073316/https://www.garykessler.net/library/file_sigs.html</a> .
<b>File signature</b>	Hex: 50 4B 07 08 ASCII: P K bel bs	At beginning of first file of a split or spanned ZIP archive. From <a href="https://web.archive.org/web/20221112073316/https://www.garykessler.net/library/file_sigs.html">https://web.archive.org/web/20221112073316/https://www.garykessler.net/library/file_sigs.html</a> .
<b>Other</b>	NF00458	See <a href="https://www.archives.gov/files/lod/dpframework/id/NF00458.ttl">https://www.archives.gov/files/lod/dpframework/id/NF00458.ttl</a> .
<b>Pronom PUID</b>	x-fmt/263	See <a href="https://www.nationalarchives.gov.uk/PRONOM/x-fmt/263">https://www.nationalarchives.gov.uk/PRONOM/x-fmt/263</a> .
<b>Wikidata Title ID</b>	Q136218	See <a href="https://www.wikidata.org/wiki/Q136218">https://www.wikidata.org/wiki/Q136218</a> .

## Notes

<b>General</b>	<p><b>Use of ZIP file format in other specifications:</b> The notes here describe the relationship of some specifications based on ZIP to versions and features of ZIP_PK. These notes are based on the specification documents.</p> <ul style="list-style-type: none"> <li>The <b>SCORM</b> (Sharable Content Object Reference Model, used for learning materials) includes a Package Interchange Format (PIF) based on ZIP. The SCORM 2004 Content Aggregation Model mandates compliance with PKZip v 2.04g and <a href="#">RFC 1951 (DEFLATE)</a> as the only compression algorithm. Numbering of versions of the PKZip software and the APPNOTE.TXT specification do not correspond; however, the timelines on <a href="https://en.wikipedia.org/wiki/PKZIP">https://en.wikipedia.org/wiki/PKZIP</a> and <a href="https://en.wikipedia.org/wiki/ZIP_(file_format)">https://en.wikipedia.org/wiki/ZIP_(file_format)</a> suggest that SCORM 2004 usage corresponds to APPNOTE.TXT version 2.x.</li> <li><b>JAR</b> files, packages used to transmit software in the Java programming language are also based on the ZIP file format. The JAR specification was originally based on an <a href="#">APPNOTE.TXT version dated 1997-03-11 modified by Info-ZIP</a> and based on a PKWARE version dated 1996-02-15. The <a href="#">JAR utility description</a> also specifies that file-names must be encoded in UTF-8. Compression, if applied, uses <a href="#">RFC 1951 (DEFLATE)</a>. Version 7 of the Java platform (current as of September 2012) refers to the latest APPNOTE.TXT for optional use of the ZIP64 extension to support large files and for indicating the use of UTF-8 (Appendix D). The Java platform does not support encryption. [Aside: program code can be <i>obfuscated</i> (e.g., by changing meaningful names for variables, etc. to "a," "b," ...) to make it hard to read.]</li> <li>The <b>Open Packaging Conventions (OPC, Part 2 of ISO/IEC 29500, OOXML)</b> are based on <a href="#">version 6.2.0 of APPNOTE.TXT</a>. Compression is restricted to DEFLATE; digital signatures are not supported. Details on the use of ZIP in OPC are in section 10 and Annex C of ISO/IEC 29500-2:2012.</li> <li>Version 1.2 of <b>ODF</b> (Open Document Format for Office Applications) from OASIS incorporates <a href="#">version 6.2.0 of APPNOTE.TXT</a>. The only compression algorithm allowed is DEFLATE (RFC 1951). ODF defines its own encryption mechanism. Version 1.0 of ODF (as published as ISO/IEC 26300:2006) referred to the <a href="#">APPNOTE.TXT version dated 1997-03-11 from Info-ZIP</a>, because no appropriate version was made available publicly by PKWARE at the time because the version 6.2.0 referred to by the original ODF specification from OASIS had been superseded. Now that PKWARE maintains an archive of documentation to facilitate persistent</li> </ul>
----------------	--

references, Version 1.2 of ODF again refers to version 6.2.0 of APPNOTE.TXT at PKWARE.

- Versions 2 and 3 of **EPUB** define OCF (Open Container Format) as a package for encapsulating the components of an electronic publication. [EPUB\\_3](#) mandates a container format based on ZIP as defined by [version 6.3.0 of APPNOTE.TXT](#). In [EPUB\\_2](#), the ZIP-based container was supported but not mandated. The only compression algorithm allowed is DEFLATE (RFC 1951). Split ZIP archives are explicitly prohibited. Encryption is only allowed using an EPUB-specific approach. File names, etc. must be in UTF-8. Use of the PKWARE ZIP64 extension (version 1 only) is allowed but discouraged unless the content requires it.

Bearing in mind changes in technology since the original ZIP\_PK, it appears to the compilers of this description that a common set of features for the most interoperable usage is emerging, including: the basic ZIP\_PK structure with CRC-32 for checking file integrity, DEFLATE compression (or no compression), and support for long filenames, for filenames, etc., encoded in UTF-8, and for large files through ZIP64. ZIP\_PK features avoided in these specifications include: encryption as supported in ZIP\_PK; other compression schemes; other features covered by PKWARE patents. Application-specific approaches have been developed for file manifests (with more information than in the ZIP Central Directory), metadata, digital signatures, and encryption. [Comments welcome.](#)

**PKWARE patents and licensing:** One set of changes between [version 6.3.2 \(2007\)](#) and [version 6.3.3 \(2012\)](#) of APPNOTE.TXT was intended to add clarity to the specification in relation to the scope and applicability of PKWARE's patents, making clearer to potential developers the particular sections of the specification that have any relationship to PKWARE's patents, with the implication that reliance on other sections should provide no patent concerns. Statements in version 6.3.3 include: "Portions of the Strong Encryption technology defined in this specification are covered under patents and pending patent applications." and "Patch support is provided by PKPatchMaker(tm) technology and is covered under U.S. Patents and Patents Pending." [See explanation of "patch technology" immediately below.] In specific sections of version 6.3.3 of APPNOTE.TXT that relate to these two features, pointers are made to Section 10 of version 6.3.3 of APPNOTE.TXT, entitled "Incorporating PKWARE Proprietary Technology into Your Product." This section includes the following statement, "The Use or Implementation in a product of APPNOTE technological components pertaining to either strong encryption or patching requires a separate, executed license agreement from PKWARE." and provides contact information. These formal statements may be read in the context of the purpose expressed in the new section 1.1.1 of version 6.3.3 of APPNOTE.TXT, "This specification is intended to define a cross-platform, interoperable file storage and transfer format. Since its first publication in 1989, PKWARE, Inc. ("PKWARE") has remained committed to ensuring the interoperability of the .ZIP file format through periodic publication and maintenance of this specification. We trust that all .ZIP compatible vendors and application developers that use and benefit from this format will share and support this commitment to interoperability."

PKWARE declares the patents it owns at <https://support.pkware.com/home/pkzip/developer-tools/appnote/intellectual-property>.

**Patching technology support in ZIP format:** Jim Peterson of PKWARE provided the following explanation: The "patching" technology is a means for using ZIP files to distribute revised document content by delivering only the changed elements of a prior document instead of having to deliver a complete new copy of the revised version. The premise is that a user delivers version 1 of a file to other users. Later, version 2 is prepared for delivery to the same recipients. The patching technology provides capabilities to compare version 1 and version 2 and distill out only the

differences between the documents. The user distributing version 2 can then deliver only the differences rather than a whole new document with the objective of reducing the amount of information to deliver to update a user from version 1 to version 2. Upon receipt, the changes can be applied directly to the original version 1 of the document/file with the end result that version 2 becomes available to the receiving users through the patching technology.

**Support for ZIP\_PK features:** The [2012 version 6.3.3 of the ZIP Application Note](#) indicates, "Newer versions of this format specification are intended to remain interoperable with all prior versions whenever technically possible." Although PKWARE products are likely to support all features in ZIP\_PK, other ZIP utilities will not necessarily support optional features. In particular, support for any particular compression or encryption method is optional. Section 4 of the 2012 version 6.3.3 of APPNOTE.TXT provides a clearer specification of which features are mandatory and which optional than earlier versions. Section 4.1.3 states, "Compression method 8 (Deflate) is the method used by default by most Zip compatible application programs." Apache Commons Compress, a software toolkit built in Java, provides some informative details on creating interoperable ZIP files at <http://commons.apache.org/proper/commons-compress/index.html>

**Variants from other vendors:** Due to differences in timing of introduction of features in different products, such as WinZip and the Info-ZIP tools, some files with a .zip extension produced in the past may not strictly comply with the ZIP\_PK specification as represented by a particular edition of APPNOTE.TXT. Others may use the mechanisms for extensibility in ZIP\_PK in ways that other utilities do not recognize. In general, a consensus towards convergence has emerged in recent years. For example, features introduced by other developers have been folded back into the ZIP\_PK specification. The compilers of this resource are unable to determine how serious the potential problem of incompatible legacy files is in practice. Experience at the Library of Congress with the use of ZIP files for transmission of digital content to the library has been almost entirely trouble-free.

The WinZip product introduced a .zipx extension to distinguish archive files that used more advanced features, retaining .zip for files using only the features supported by most ZIP utilities. Version 6.3.3 of APPNOTE.TXT mentions .zipx as an extension that may be used for files compliant with the format it specifies. The compilers of this resource are not aware of any use of the .zipx extension for files created by tools other than WinZip.  
[Comments welcome.](#)

Several different tools that have "zip" in their names have their own native storage format and do not necessarily support the ZIP\_PK format described here, for example, *gzip*; *7-zip*; and *bzip2*.

**Data integrity checks:** A mandatory feature of the ZIP\_PK format is the use of a 32-bit cyclic redundancy check (CRC) for each file using a specific CRC-32 algorithm. On decompression, ZIP utilities are expected to recalculate a CRC on the decompressed file and check against the CRC stored in the ZIP archive file. In addition, ZIP\_PK provides a way to store optional digital signatures.

## History

**Chronology for APPNOTE.TXT versions:** Since the first publication of APPNOTE.TXT in 1990, ZIP\_PK has been extended in response to a changing technological environment, while retaining its straightforward structure. As new features have been added to ZIP\_PK, care has been taken to ensure backwards compatibility, so that new software versions can read files in all previous versions of the format specification. A chronology of significant versions of the APPNOTE specification for ZIP\_PK, prior to and including 2012, follows below. The details and descriptions were provided by Jim Peterson, Chief Scientist, PKWARE. Versions not on this list introduced only minor changes, such as corrections or support for an additional compression method.



- 1.0, 1990-03-15 [<http://hdl.loc.gov/loc.gdc/digformat.000009.1>] Initial ZIP format created by Phil Katz of PKWARE, Inc. Includes original 96-bit (triple 32-bit) password encryption storage definition.
- 2.0, 1993-02-01 [<http://hdl.loc.gov/loc.gdc/digformat.000010.1>] Introduces the Deflate data compression algorithm which becomes one of the most widely used compression algorithms.
- 4.0, 2000-11-01 [<http://hdl.loc.gov/loc.gdc/digformat.000011.1>] Introduces support for using digital signatures to verify data in a ZIP file and adds the fast Deflate64 data compression algorithm.
- 4.5, 2001-11-01 [<http://hdl.loc.gov/loc.gdc/digformat.000012.1>] Introduces support for storing more than 65535 files in a ZIP file and file sizes exceeding the previous 4 gigabyte limit.
- 5.2, 2003-07-16 [<http://hdl.loc.gov/loc.gdc/digformat.000013.1>] Introduces initial storage specification for adding stronger encryption using algorithms exceeding 128-bits based on passwords, digital certificates, or a combination of both simultaneously.
- 6.2.0, 2004-04-26 [<http://hdl.loc.gov/loc.gdc/digformat.000014.1>] Defines storage specification to support encrypting ZIP file metadata, such as file names, within the ZIP Central Directory. Reference version for incorporating the ZIP format into Office Open XML File Format standard ECMA-376 (versions 1, 2, and 3) and ISO/IEC 29500 (versions published in 2008 and 2012).
- 6.2.2, 2006-01-06 [<http://hdl.loc.gov/loc.gdc/digformat.000015.1>] Documents final storage specification for adding stronger encryption using algorithms exceeding 128-bits using passwords, digital certificates, or a combination of both simultaneously.
- 6.3.0, 2006-09-29 [<http://hdl.loc.gov/loc.gdc/digformat.000016.1>] Documents support within ZIP files for using hash algorithms of SHA-256, SHA-384 and SHA-512; incorporates LZMA and PPMd data compression algorithms; support for Blowfish and Twofish data encryption algorithms; definition of UTF-8 international file name storage.
- 6.3.3, 2012-09-01 [<http://hdl.loc.gov/loc.gdc/digformat.000017.1>] Formatting changes to the document add conformance requirements and support easier referencing by other standards that incorporate the ZIP format.

**ZIP\_PK in the international standards environment:** As described in [https://en.wikipedia.org/wiki/ZIP\\_\(file\\_format\)](https://en.wikipedia.org/wiki/ZIP_(file_format)), a proposed project to create an ISO/IEC international standard for a format compatible with ZIP failed to pass a 2010 ballot of national standards bodies. Instead, a study period was initiated, resulting in recommendations documented in [ISO/IEC JTC 1/SC 34 N 1621](https://www.iso.org/standard/54441.html). The recommendations were (a) to have PKWARE continue its maintenance of the ZIP Application Note, (b) to plan for a new multi-part ISO standard to build on top of the ZIP Application Note, and (c) to propose a new work item for Part 1 of the new standard for a Document Container File. The new work item was approved in August 2011 and the standard was published in 2015, as [ISO/IEC 21320-1:2015 - Information technology -- Document Container File -- Part 1: Core](https://www.iso.org/standard/64441.html).

## Format specifications

- PKWARE provides online access to the latest specification, known as the .ZIP Application Note, and stored as APPNOTE.TXT, and an archived selection of earlier versions.
  - [Introduction to .ZIP Application Note](https://support.pkware.com/home/pkzip/developer-tools/appnote) (<https://support.pkware.com/home/pkzip/developer-tools/appnote>).
  - [Current version of APPNOTE.TXT from PKWARE](https://www.pkware.com/documents/casestudies/APPNOTE.TXT) (<https://www.pkware.com/documents/casestudies/APPNOTE.TXT>). This URL always leads to the latest version.
  - [Archive of .ZIP Application Notes from PKWARE](https://support.pkware.com/home/pkzip/developer-tools/appnote/application-note-archives) (<https://support.pkware.com/home/pkzip/developer-tools/appnote/application-note-archives>).

## Useful references

### URLs

- [Registration of internet media type for ZIP at IANA](https://www.iana.org/assignments/media-types/application/zip). (<https://www.iana.org/assignments/media-types/application/zip>).
- [Wikipedia page for ZIP file format](https://en.wikipedia.org/wiki/ZIP_(file_format)) ([https://en.wikipedia.org/wiki/ZIP\\_\(file\\_format\)](https://en.wikipedia.org/wiki/ZIP_(file_format))). Includes a version history listing key advances in the ZIP\_PK specification through versions of the APPNOTE.TXT document. Also discusses efforts since 2012 to bring the ZIP format into the international standards environment.
- ISO/IEC 21320-1 was a standards effort approved in August 2011 under ISO/IEC JTC 1/SC 34/WG1 [Markup Languages] for core requirements for a document container file.
  - [ISO catalogue entry for ISO/IEC 21320-1:2015](https://www.iso.org/standard/60101.html). (<https://www.iso.org/standard/60101.html>).
  - [ISO/IEC JTC 1/SC 34 N 1621: Proposal for a New Work Item on Information Technology -- Document Container File -- Part 1: Core](https://web.archive.org/web/20150317134106/http://kikaku.itscj.ipsj.or.jp/sc34/open/1621.pdf) (<https://web.archive.org/web/20150317134106/http://kikaku.itscj.ipsj.or.jp/sc34/open/1621.pdf>). This document provided the starting background for ISO/IEC 21320-1.
- [RFC 1951: DEFLATE Compressed Data Format Specification version 1.3](https://www.ietf.org/rfc/rfc1951.txt) (<https://www.ietf.org/rfc/rfc1951.txt>). No changes to the format had been made since version 1.1, only changes to the document. States, "Phil Katz designed the deflate format. Jean-Loup Gailly and Mark Adler wrote the related software described in this specification."
- [Wikipedia page for Cyclic Redundancy Check \(CRC\)](https://en.wikipedia.org/wiki/Cyclic_redundancy_check) ([https://en.wikipedia.org/wiki/Cyclic\\_redundancy\\_check](https://en.wikipedia.org/wiki/Cyclic_redundancy_check)). Background on CRC. ZIP\_PK uses a specific 32-bit CRC algorithm, cited in APPNOTE.TXT.
- Info-ZIP is the open-source project of a group originally formed in 1990 as a mailing list. To quote from the group's website, "Info-ZIP's purpose is to provide free, portable, high-quality versions of the Zip and UnZip compressor-archiver utilities that are compatible with the DOS-based PKZIP by PKWARE, Inc." Info-ZIP tools and libraries support Unix, Mac OS, Windows, and many other operating systems. Code from Info-ZIP has been the basis for many other ZIP implementations. Info-ZIP's documentation of the ZIP format has been relied on historically by other developers and in other specifications, such as JAR, since PKWARE made only the most recent version of APPNOTE.TXT available.
  - [Info\\_ZIP home page](http://infozip.sourceforge.net/) (<http://infozip.sourceforge.net/>).
  - [Documentation from Info-ZIP](ftp://ftp.info-zip.org/pub/infozip/doc/) (<ftp://ftp.info-zip.org/pub/infozip/doc/>). Includes Info-ZIP variants on APPNOTE.TXT dated from 1996 to 2004.
  - [Wikipedia page for Info-ZIP](https://en.wikipedia.org/wiki/Info-ZIP) (<https://en.wikipedia.org/wiki/Info-ZIP>). Includes chronology of versions of the Info-ZIP utilities.
- Resources illustrating support for ZIP format in mobile operating systems, iOS and Android
  - [How to Open Zip Files on an iPhone or iPad \(updated November 2018\)](https://www.howtogeek.com/212703/how-to-open-zip-files-on-an-iphone-or-ipad/). (<https://www.howtogeek.com/212703/how-to-open-zip-files-on-an-iphone-or-ipad/>).
  - [How to Unzip a Zip File With Your iPhone or iPad's Files App \(November 2018\)](https://www.howtogeek.com/fyi/how-to-unzip-a-zip-file-with-your-iphone-or-ipads-files-app/). (<https://www.howtogeek.com/fyi/how-to-unzip-a-zip-file-with-your-iphone-or-ipads-files-app/>).
  - [5 best zip, rar, and unzip apps for Android \(May 2020\)](https://www.androidauthority.com/the-best-zip-and-unzip-apps-for-android-258692/) (<https://www.androidauthority.com/the-best-zip-and-unzip-apps-for-android-258692/>).
  - [10 best Android file explorer apps, file browsers, and file managers! \(November 2019\)](https://www.androidauthority.com/file-manager-explorer-apps-android-279800/) (<https://www.androidauthority.com/file-manager-explorer-apps-android-279800/>). Some file managers for Android will open ZIP files and allow extraction of files.
- [Java Archive \(JAR\) Files, Java SE, version 7](https://docs.oracle.com/javase/7/docs/technotes/guides/jar/index.html) (<https://docs.oracle.com/javase/7/docs/technotes/guides/jar/index.html>). The Java programming language was originally developed at Sun Microsystems, acquired in early 2010 by Oracle. The associated toolkit includes java.jar.util, which, in turn, depends on java.zip.util.
- [ZIP Package in Apache Commons Compress toolkit](https://commons.apache.org/proper/commons-compress/zip.html) (<https://commons.apache.org/proper/commons-compress/zip.html>). This page describes how this package extends the java.zip.util that it is based on. It also has useful tips on creating interoperable ZIP files.
- [Third-party ZIP and GZIP modules for JHOVE, from Portico](http://jhove.sourceforge.net/thirdpartymodules/zipmodule.html) (<http://jhove.sourceforge.net/thirdpartymodules/zipmodule.html>). Retrieved 2012-09-22. Described as draft and dated 4-April-2007.
- [PKWARE, Inc., Third-Party License Summary for ZIP Reader \(2012\)| link via Internet Archive](https://web.archive.org/web/20120429034552/http://www.pkware.com/Support-Desktop/zipreaderthirdparty) (<https://web.archive.org/web/20120429034552/http://www.pkware.com/Support-Desktop/zipreaderthirdparty>).
- [WinZip: AES Encryption Information](https://www.winzip.com/aes_info.htm) ([https://www.winzip.com/aes\\_info.htm](https://www.winzip.com/aes_info.htm)). This page (last updated in 2009) discusses how WinZip extended the ZIP format to support AES encryption. Although the page does not mention PKWARE, the phrase "basic ZIP file format" refers to a modified version of APPNOTE, distributed by Info-Zip, based on PKWARE's APPNOTE dated Feb 15, 1996, and updated to take account of the 6.2.0 version dated April 26, 2004.
- [Is WinZip compatible with older versions of WinZip and other Zip file utilities?](https://support.winzip.com/hc/en-us/articles/115011349147-Is-WinZip-compatible-with-older-versions-of-WinZip-and-other-Zip-file-utilities-) (<https://support.winzip.com/hc/en-us/articles/115011349147-Is-WinZip-compatible-with-older-versions-of-WinZip-and-other-Zip-file-utilities->). Explains WinZip options for creating .zip or .zipx files. The .zip files created by WinZip are intended for greatest compatibility with other Zip file utilities.

- [NARA File Format Preservation Plan ID entry for NF00458](https://www.archives.gov/files/lod/dpframework/id/NF00458.ttl) (https://www.archives.gov/files/lod/dpframework/id/NF00458.ttl). Information in NARA File Format Preservation Plan ID about ZIP archive.
- [ZIP file format entry at PRONOM](https://www.nationalarchives.gov.uk/PRONOM/x-fmt/263) (https://www.nationalarchives.gov.uk/PRONOM/x-fmt/263). PUID is x-fmt/263
- [Wikidata entry for Q136218](https://www.wikidata.org/wiki/Q136218) (https://www.wikidata.org/wiki/Q136218). Information in Wikidata for ZIP family of archive file formats. Wikidata Title ID: Q136218

Last Updated: 05/17/2024

[Digital Preservation Home](#) | [Digital Formats Home](#)