Huge documents in LibreOffice (Zip64 support)

Attila Szűcs

Software Engineer

Attila.szucs@collabora.com

Collabora Online

LibreOffice Conference Bucharest 2023

LibreOffice Technology

What is ZIP64 and why it is needed

- Most document formats are compressed with zip
- The original .ZIP file format was designed at year 1989
  old system – old limitations
  e.g. filesize stored in 32bit = max 4gb
  extensible
- technology advanced -> limitations reached
- Year 2001: ZIP64 extension

Collabora Online

ZIP64 new limitations

1) uncompressed file size: 32bit -> 64bit (4gigaB->16exaB)
2) compressed archive size: 32bit -> 64bit
3) file count: 16bit -> 32bit (64k->4g)
4) disk count: 16bit -> 32bit
5) Some less important internal limitations
   like size of the central directory

For most LibreOffice Documents, old ZIP limitations are enough, practically only uncompressed filesize can be problem for a while.

Where are these data stored in the zip archive?

Original ZIP format

ZIP archive built from smaller parts (Records)
Some of the record start with a signature

local file header 1
file data 1
[data descriptor 1]
...
local file header n
file data n
[data descriptor n]
...
Central directory

File header 1
...
File header n
...
End of central directory record

New/extended records to store data

Extended records

Implemented export/import in LibreOffice

- Local file header
- File header (In Central directory)
- Data descriptor

Extra field

New records

Not implemented in LibreOffice

- zip64 end of central directory record
- zip64 end of central directory locator

ZIP64 is not 1 property for the entire archive.

Every record can be independently in ZIP64 mode or not.

Title of the slide

Local file header:

	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	0xe	0xf
0x000	Signature				Version		Flags		Compr ession		Mod time		Mod Date		Crc-32	
0x010	Crc-32		**Compressed Size**				**Uncompressed Size**				File Name len		**Extra field len**			
0x020	File Name (variable size)															
0x030	**Extra field** (variable size)															

If compressed size, (or/and?) Uncompressed size == 0xFFFFFFFF
then the real value in the extra field

Extra field:

	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	0xe	0xf
0x000	ID		Size		Extra data (variable size)											
0x010	1		28*		**Uncompressed size**								**Compressed-**			
0x020	**-size**				Relative offset header								Disk start number			

General extra field

Zip64

*Can be smaller. For example: 16. The Zip Standard does not mention or forbid it.

Data descriptor

Rarely used. Designed for file streaming.

- Signature – 4 byte — Not in the standard but commonly used.
- Crc-32 – 4 byte
- Compressed size – 4 byte — 8 byte in case of Zip64
- Uncompressed size -4 byte

The standard

- Designed to be well expandable
- Allow a lot of things.. even senseless things
- Commonly used many extension
- Complex, many special cases

  (encrypted, compressed, streamed, split, self extract, ...)
- Not exact

  "4.3.9.3 Although not originally assigned a signature, the value 0x08074b50 has commonly been adopted as a signature value for the data descriptor record."
- Hard to prepare for every use case

Collabora Online

Test case

Unittest: small zip64 files. (fast to load)

Manual test for 4gb+ (content.xml) size (works but slow)
it is a challenge to create the testfile.

Release: several minutes to import.
Debug: it was like 40 minutes for me.

Future possibilities

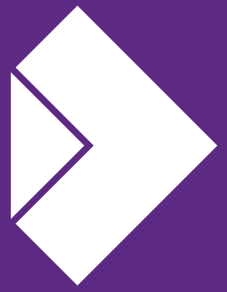
Compressed size 32bit→64bit (Partially implemented)
What need:

- load/save zip64 end of central directory record / locator
- Make sure all usage is 64bit compatible

  1 local sal_uInt32 variable, function parameter, or return value can break everything

Some code pointers:

- sal_Int32 ZipFile::readCEN()
- void ZipOutputStream::writeCEN(const ZipEntry &rEntry)
- sal_Int32 Deflater::doDeflateBytes
- void ZipFile::recover()



Collabora Online

Thank you!

By Attila Szűcs

@CollaboraOffice
hello@collaboraoffice.com
www.collaboraoffice.com