

Technical Note – ZIP File Capacities

Date issued:	08-01-2014	Notice sent to:	PKWARE Technical Support, Sales Engineers; Customer's, Prospects
Effective date:	Immediately	Products affected:	<ul style="list-style-type: none">• PKZIP/SecureZIP All Platforms
Technical Support Ticket ID:	N/A	Location of completed fix:	N/A
Author:	JPeterson		

Short description of issue: This note documents a number of important ZIP file capacities such as how large a ZIP file can be, how many files it can contain, and other storage related limits applicable to .ZIP files. This information affects PKSFX capacities also. These capacities affect PKZIP and SecureZIP as well as all other ZIP compatible applications that conform to the published specification for .ZIP files.

The specification for the .ZIP file format has been publicly available and distributed by PKWARE in a file called APPNOTE.TXT. The APPNOTE documents the internal data structures and layout that define a .ZIP archive. The current APPNOTE specification is available from PKWARE at <http://www.pkware.com/appnote.html>.

Detailed description of issue:

The original .ZIP file format met the needs of computer users at the time it was introduced by PKWARE in 1989. As computer technology advanced over time, storage capacities increased dramatically. These increases make the numbers and sizes of files that seemed unimaginable back in 1989 a reality today. To extend the utility of the .ZIP file format to meet these changing system needs, PKWARE has continued to extend the .ZIP file format to support more than 65,535 files per archive and archive sizes greater than 4 Gigabytes (GB). The extensions introduced by PKWARE fully support all the features of existing archives. Versions of PKZIP and SecureZIP that support these extensions will continue to read all existing archives.

Versions of PKZIP that supported the version 4.5 APPNOTE specification, or earlier, were limited to storing no more than 65,535 files in a .ZIP archive. The original version of PKZIP for DOS v2.04g matching to the 2.0 APPNOTE could store no more than 16,383 files per .ZIP archive.

Another limitation that existed prior to the 4.5 version of PKZIP was that a single .ZIP archive could not be larger than 4 GB (4,294,967,295 bytes). The original version of PKZIP for DOS cannot process .ZIP archives larger than 2 GB (2,147,483,647 bytes).

Version 4.5 of the .ZIP file format specification supports creating ZIP archives containing over 4 billion files and having sizes larger than 9 quintillion bytes (Exabytes). These are only theoretical limits based on the defined storage parameters in the ZIP format. Most computer systems in common use today do not have enough storage capacity or available memory to create and store .ZIP archives approaching these limits.

The practical limits imposed by a typical computer in use today and configured with sufficient memory will support compressing up to approximately 262,144 files. Compressing this number of files can take a long time due to the time required to process this number of files. Since it is not practical to reach the theoretical limits supported by the extended .ZIP file format, PKZIP 4.5 or later, will currently not compress or extract more than 2,147,483,647 files.

The limits on the numbers of files and sizes of archives can vary depending on the operating system you are using. The actual numbers may that may be achieved by a customer when creating a .ZIP files will vary depending on available memory and storage resources on the system in use.

The table below lists the capacities that are currently supported by the ZIP format.

Capacities	Current Version
Maximum ZIP archive size	9,223,372,036,854,775,807 bytes
Maximum number of files in archive	2,147,483,647 files
Number of segments for spanned and split ZIP archives	4,294,967,295 segments
Maximum size of segments for spanned and split ZIP archives	4,294,967,295 bytes
Maximum stream size for Application Integration*	4,294,967,294 bytes
Maximum file comment size	65,535 bytes
Maximum number of passwords per file or per archive	1
Maximum encryption recipients**	3275
Maximum signatures per file***	200
Maximum signatures on a central directory	1
Maximum PKSFX size (not split)	2,147,483,647 bytes
Maximum PKSFX split segment size	2,147,483,647 bytes

**This limitation applies to the use of streamed data on Windows, UNIX and Linux platforms only. This limitation is expected to be removed in the future.*

***Storage for encryption certificate data cannot exceed 64K. The number of recipients that can be stored will vary based on size of certificates. Approximately 20 bytes is required to store information for each certificate.*

****Storage for signing certificate data cannot exceed 64K. The number of signatures that can be stored will vary based on size of certificates. Approximately 300 bytes is required to store information for each certificate.*

PKSFX® self-extracting file sizes are limited to the size of an executable program file supported by the underlying operating system. This currently is 2 GB for most operating systems. PKZIP can be configured to create PKSFX files larger than this limit however they will not run on most operating systems. To create and run PKSFX files that have the capacity to hold more than 2 GB create a split PKSFX file where each segment is not larger than 2 GB. Another important note for PKSFX files is that when creating a split archive, the segment size should not be set smaller than the extraction component for the target platform. This means the extraction engine cannot be split and must be fully contained within a segment.

When creating archives that take advantage of these larger capacities, you should be aware that much older versions of PKZIP as well as ZIP programs from some other vendors will not be able to recognize all of the files in the new archives you create that exceed the capacities of the older programs. If you plan to send a large archive to a friend or associate, they will need a compatible version of PKZIP in order to extract the contents of the file(s) you send. To ensure archives can always be read, always use genuine PKZIP/SecureZIP.