

File Compression and Decompression

The NTFS file system volumes support file compression on an individual file basis. The file compression algorithm used by the NTFS file system is Lempel-Ziv compression. This is a *lossless* compression algorithm, which means that no data is lost when compressing and decompressing the file, as opposed to *lossy* compression algorithms such as JPEG, where some data is lost each time data compression and decompression occur.

Data compression reduces the size of a file by minimizing redundant data. In a text file, redundant data can be frequently occurring characters, such as the space character, or common vowels, such as the letters e and a; it can also be frequently occurring character strings. Data compression creates a compressed version of a file by minimizing this redundant data.

Each type of data-compression algorithm minimizes redundant data in a unique manner. For example, the *Huffman encoding algorithm* assigns a code to characters in a file based on how frequently those characters occur. Another compression algorithm, called *run-length encoding*, generates a two-part value for repeated characters: the first part specifies the number of times the character is repeated, and the second part identifies the character. Another compression algorithm, known as the *Lempel-Ziv algorithm*, converts variable-length strings into fixed-length codes that consume less space than the original strings.

The NTFS File System File Compression

On the NTFS file system, compression is performed transparently. This means it can be used without requiring changes to existing applications. The compressed bytes of the file are not accessible to applications; they see only the uncompressed data. Therefore, applications that open a compressed file can operate on it as if it were not compressed. However, these files cannot be copied to another file system.

If you compress a file that is larger than 30 gigabytes, the compression may not succeed.

The following topics identify the NTFS file system file compression:

- [Compression Attribute](#)
- [Compression State](#)
- [Obtaining the Size of a Compressed File](#)

File Compression and Decompression Libraries

The file compression and decompression libraries take an existing file or files and produce a file or files that are compressed versions of the originals. The compression is also lossless, but the compression is not transparent to applications. An application can only operate on such files with the assistance of a file compression library. In addition, the only operations you can perform on such files are creating a compressed file from an original and recovering the original data from the decompressed version. Editing is typically not supported, and seeking is limited if supported at all.

Typically, an application calls functions in Lz32.dll to decompress data that was compressed using Compress.exe. The functions can also process files without attempting to decompress them.

You can use the functions in Lz32.dll to decompress single or multiple files. You can also use them to decompress compressed files a portion at a time.

The following topics identify the file decompression that is provided by the functions in Lz32.dll:

- [Decompressing a single file](#)
- [Decompressing multiple files](#)
- [Reading from compressed files](#)

Cabinets

Cabinets are created by a compression library that supports features such as disk spanning and multi-file compression. For additional information, see the Cabinet Software Development Kit: <http://msdn.microsoft.com/library/en-us/dncabsdk/html/cabdl.asp>.

In this section

Topic	Description
Compression Attribute	On an NTFS file system volume, each file and directory has a <i>compression attribute</i> .
Compression State	Each file and directory on a volume that supports compression for individual files and directories has a <i>compression state</i> .

Obtaining the Size of a Compressed File

To obtain the compressed size of a file use the GetCompressedFileSize function.

Decompressing a Single File

An application can decompress a single compressed file by using the LZOpenFile, LZCopy, and LZClose functions.

Decompressing Multiple Files

An application can decompress multiple files by using the LZOpenFile, LZCopy, and LZClose functions.

Reading from Compressed Files

An application can decompress a compressed file a portion at a time by using the LZSeek and LZRead functions.

Community Additions

© 2014 Microsoft. All rights reserved.