

Microsoft MSZIP Data Compression Format

Copyright © 1997 Microsoft Corporation. All rights reserved.

Abstract

This document explains how to encode or decode MSZIP data compression format, as used in Microsoft cabinet files, using publicly-available code from the zlib library. This information may be used to create or extract Microsoft cabinet files which utilize MSZIP compression. The format of a cabinet file is described in other documents.

Introduction

This document describes the format of MSZIP compressed data as used in the MSZIP compression mode of Microsoft's cabinet files. The purpose of this document is to allow anyone to encode or decode MSZIP compressed data.

Implementation Details

MSZIP compression has only minor variations from Phil Katz's 'deflate' method. Rather than re-document this method, this document will explain these variations and refer the reader to publicly-available 'deflate' documents. Some 'deflate' implementations may contain extensions to the original specifications, but MSZIP uses only the three basic modes of deflate: stored, fixed Huffman tree, and dynamic Huffman tree.

Each MSZIP data block is the result of a complete 'deflate' compression operation. Each block is flushed out of the compressor before the next block begins, so the last sub-block in each block will be marked as the 'end' of the stream. Any decoding trees are discarded after each block, with only the history buffer surviving from one block to the next. Each data block represents 32k uncompressed, except that the last block in a folder may be smaller. A two-byte MSZIP signature precedes the compressed encoding in each block, consisting of the bytes 0x43, 0x4B.

The maximum compressed size of each MSZIP block is 32k + 12 bytes. This allows for the data to be passed as two separate "stored" sub-blocks, which each have a 5-byte overhead, plus the 2-byte signature. The Microsoft MSZIP compressor will emit "stored" sub-blocks with a length of exactly 32k, while some implementations do not exceed 32k-1.

Whenever a cabinet folder boundary is reached, the compression history is discarded, so that decoding any folder does not require any prior data.

Where to find the 'deflate' Specifications

The 'deflate' algorithm was original documented by Phil Katz in APPNOTE.TXT, which accompanied the PKZip software. It's most-recent description can be found in RFC 1951. (Try <ftp://ftp.uu.net/graphics/png/documents/zlib/zdoc-index.html> for pointers to obtain this RFC.)