

With PCDEZIP you can confidently download

files compressed with PKZIP, Version 2.0.

**BY BOB FLANDERS AND
MICHAEL HOLMES**

Decompress files with PCDEZIP

If you've been using PCUNZIP to decompress files, it's time to upgrade to the new PCDEZIP. Phil Katz has released Version 2.0 of his popular shareware compression program, PKZIP, and the old utility can't decompress the newly zipped files. The new PCDEZIP will unzip files zipped with either version of PKZIP.

Although we have added some valuable new features to PCDEZIP, our attempt has not been to duplicate all the capabilities of the commercial product. PCDEZIP does not, for example, permit using wildcard characters in the .ZIP filename, handle files that span more than one floppy disk, preserve read-only file attributes, or decrypt encrypted files. To get these facilities — and, of course, to be able to zip files yourself — you must obtain the full-scale PKZIP, which is available from My Computer Company on (02) 565 1991. Our aim here is simply to provide our readers with a no-cost way to decompress MagDisk files.

You can get a copy of PCDEZIP by downloading PCDEZIP.EXE from MagDisk, as explained in the sidebar 'How to obtain a copy of PCDEZIP'.

After downloading, you should rename the file PCDEZIP.EXE. To study or change the source code, download the self-extracting archive file PCDZ.EXE.

After running PCDZ.EXE, you can use the included C.BAT batch file to compile the program. The parameter you supply to C.BAT specifies the compiler you're using. To compile with Borland C++ 3.1, Borland Turbo C++ 3.0, or Microsoft C/C++ 7.0 (or later), use the parameters **C B**, **C T**, or **C M**,

PCDEZIP's syntax

```
usage: PCDEZIP [switches] zipfile [target\] [filespec [...]]

switches: -f  freshen existing files
          -n  extract new and updated files
          -o  do not prompt on overwrite
          -v  view ZIP directory
          -d  create directories
          -t  test file integrity

zipfile   is source .ZIP file

target\   is the target directory; trailing backslash required

filenames are files to be extracted; wildcards supported
```

Fig 1: When you enter PCDEZIP with no parameters, it displays its syntax and switches as shown here

respectively. If you don't supply a compiler parameter, C.BAT assumes Borland C++ 3.1.

Using PCDEZIP

The full command line syntax for PCDEZIP is

```
PCDEZIP [switches] zip_file  
[target\] [filespec [. . .]]
```

Entered without any of its optional parameters, the command

```
PCDEZIP zip_file
```

simply decompresses the contents of the named .ZIP file and places them into your current directory. You do not need to enter the .ZIP extension; PCDEZIP assumes it unless you specify otherwise.

If the .ZIP file is not in your current directory, you have to provide the path to

it. For example, if SOURCE.ZIP is in your D:\PROJECT directory, then the command

```
PCDEZIP D:\PROJECT\SOURCE
```

will expand SOURCE.ZIP and put its files into your current directory.

You can select a different target directory by using PCDEZIP's *target* option. To specify a target directory, you type in the directory pathname as the argument directly following the .ZIP file to be processed. Note that the target argument must end with a backslash, because this is how PCDEZIP identifies it as a directory. Thus, to put the decompressed contents of the SOURCE.ZIP file shown above into E:\JOB, you would enter

```
PCDEZIP D:\PROJECT\SOURCE E:\JOB\
```

If the target directory you desire does not already exist, you can instruct PCDEZIP

to create one for you, unless the .ZIP file already contains its own directory information. (.ZIP files that do contain such information are created with the -apr option of PKZIP and are considered below in the section, 'Restoring directories'). To do this, you make use of the /D switch with the *target* option. Assuming that SOURCE.ZIP is an ordinary .ZIP file that contains no stored directory information and that you want to create the \JOB directory on drive E:, the command would be

```
PCDEZIP /D D:\PROJECT\SOURCE
E:\JOB\
```

The six optional switches that PCDEZIP supports are listed and briefly explained in Figure 1. This figure shows the help message that is displayed when you enter PCDEZIP without any arguments. The command line switch may actually be placed either before or after the *zip_file* entry. Like the command line switches of other APC utilities, the switches for PCDEZIP are identified by a slash (/). For readers more accustomed to the syntax of the commercial PKUNZIP, a hyphen (-) may be used in place of the slash, except

when the switch is placed after *zip_file*. Only the slash can be used if the switch parameter follows the name of the file to be decompressed.

Three of the PCDEZIP switches, /F, /N, and /O, allow you to eliminate the utility's overwrite prompt. Before PCDEZIP decompresses a zipped file, it checks to see whether there is a same-named file in the destination directory. If so, PCDEZIP normally displays the filename, the DOS directory information for both files, and the message, 'Overwrite existing file? (Y/n/all/stop)'. Press Y to replace the file, N to skip that file, A to overwrite both it and all subsequent files without further prompting, and S to stop the utility and return to DOS.

The /F switch freshens existing files. It decompresses *only* those files that have an older, same-named version in the destination directory. When the /F is specified, any same-named but earlier-dated files are automatically overwritten. The /N (newer) switch unzips the same files as /F but also adds any files that do *not* have a same-named version in the destination directory. Specifying this switch will not overwrite any later-dated, same-named files, however.

The /O switch tells PCDEZIP to overwrite any existing same-named files without regard for their date and without prompting for confirmation. Use this switch with care!

You can test the integrity of a .ZIP file with the /T (test) switch. When you specify the test switch, PCDEZIP performs a complete decompression operation on each requested file but creates no output files. If an error is detected, PCDEZIP displays an error message next to the file that contains the error.

You can view the contents of a .ZIP file without decompressing it by employing the /V switch. Frequently, your reason for wanting to use this switch is that you wish to unzip only one or two specific files from the .ZIP files. This is done by using PCDEZIP's *filespec* option.

Suppose, for example, that you have a .ZIP file named DATA.ZIP in the root directory and you enter the command

```
PCDEZIP /V \DATA
```

Figure 2 shows the hypothetical contents of this file in the same way that /V displays them. Notice that in this instance, the .ZIP file contains compressed files and directory information too. Specifically, the INCOME.DAT and INCOME.IDX files are prefaced with directory names, although INDEX.DAT stands alone. If your interest is in decompressing only INDEX.DAT, you would use the *filespec* argument to specify the file you wanted by entering the command

```
PCDEZIP \DATA INDEX.DAT
```

Note that as shown in the syntax line above, the *filespec* argument may be used to specify more than one file. This option also supports the use of wildcards.

Restoring directories

When creating a .ZIP file, PKZIP can optionally store the pathname along with the filename. In Figure 2, PKZIP used the directory tree seen in Figure 3 to put all the files in all of the subdirectories below the \ACCT directory into \DATA.ZIP.

Preserving the directory structure stored in a .ZIP file while expanding its files is the other of the two jobs of PCDEZIP's /D switch. To restore the contents and structure of the \ACCT directory tree, you would use the commands

```
CD \ACCT
PCDEZIP /D \DATA
```

If the /D switch is specified when PCDEZIP searches through the .ZIP file, it creates any directories whose names have been stored as part of the compressed files.

Thus, when PCDEZIP encounters 1991\INCOME.DAT, it looks first to see whether the 1991 directory exists in the current (\ACCT) directory. If not, PCDEZIP creates such a directory before restoring 1991\INCOME.DAT.

If you don't specify the /D switch, PCDEZIP will, by default, attempt to put every file in \DATA.ZIP into your current directory. This could be a problem. For example, if you issued the command

```
PCDEZIP \DATA
```

PCDEZIP will place INCOME.DAT and INCOME.IDX into the current directory by decompressing the contents of 1991\INCOME.DAT and 1991\INCOME.IDX, respectively. When PCDEZIP then finds INCOME.DAT and INCOME.IDX in the .ZIP file's \1992 directory, however, it will again try to decompress the files into the current directory.

Since same-named files are now found there and you did not specify the /O switch, PCDEZIP will ask whether you want to overwrite these just-decompressed files. The same thing will happen with the \1993 files. This problem is avoided by using the /D switch to recreate the original directory structure that was stored in the .ZIP file by using PKZIP's -apr option.

The filespec option can be combined with the /D switch. Thus, if you wanted to decompress all the .DAT files and restore them in their original directories, you would use the commands

```
CD \ACCT
PCDEZIP /D \DATA *.DAT
```

This will unzip 1991\INCOME.DAT, 1992\INCOME.DAT, 1993\INCOME.DAT, and INDEX.DAT. Since the /D switch was specified, the associated directories will be created as necessary. Further, since /O was not specified, PCDEZIP will prompt before overwriting any existing file.

The filespec argument can also specify the directory portion of the file desired. For example, to unzip 1991\INCOME.DAT, you would enter the following:

```
PCDEZIP /D \DATA 1991\INCOME.DAT
```

Again, the presence of the /D switch tells PCDEZIP to create the 1991 directory that contains INCOME.DAT. If you don't use this switch, PCDEZIP will put INCOME.DAT into your current directory.

As with DOS commands, however, PCDEZIP cannot accept wildcards in the path portion of the filespec argument. For example, you would unzip all of the files in the \1992 directory (creating that directory if necessary) with the command

```
PCDEZIP /D \DATA 1992\*.*
```

If you wanted to unzip all the files in only the 1991, 1992, and 1993 directories, however, you might be tempted to use the command

```
PCDEZIP /D \DATA 199?\*.*
```

This attempt would fail, however, because you cannot specify a wildcard as part of a subdirectory definition. Since PCDEZIP does not decode wildcards used in the path portion of the filename arguments, it will not match 199? with any of the directories. To unzip the desired files, you must instead use the command

```
PCDEZIP /D
\DATA 1991\*.* 1992\*.* 1993\*.*
```

Empty directory processing

When PKZIP builds a .ZIP file, it adds entries that note the existence of empty directories. This feature is needed to allow the complete restoration of a file system or subdirectory tree. Although an empty directory may seem useless, it sometimes serves as an important placeholder for temporary files that are deleted when a program finishes executing.

As indicated above, when you specify the /D switch, PCDEZIP recreates the directory structure contained in the .ZIP file. However, PCDEZIP creates directories only as needed. If you have a .ZIP file named TEMP.ZIP containing DIR1\INFO.DAT and DIR2\NOTES.TXT and issue the command

```
PCDEZIP /D TEMP
```

PCDEZIP will create both the \DIR1 and \DIR2 directories to hold INFO.DAT and NOTES.TXT, respectively. If instead you enter

```
PCDEZIP /D TEMP *.TXT
```

however, PCDEZIP will create only \DIR2, since that is what is needed to hold NOTES.TXT. The \DIR1 directory would not be created because INFO.DAT was not unzipped from the .ZIP file.

This would seem to preclude the creation of empty directories, since PCDEZIP never has a need to create them. You can, however, force PCDEZIP to create empty directories by using *.* as the filespec. Moreover, if you do not specify a filespec argument, PCDEZIP uses the *.* specification by default.

Inside PCDEZIP

Fortunately, there's no need to enter into the monumental complexities of data-compression schemes. From PCDEZIP's point of view, how the data gets encoded is of little importance: PCDEZIP simply wants to return the contents of the .ZIP file in useable form. To do so, PCDEZIP must understand the .ZIP file format, scan the .ZIP file to find the files that must be decompressed, and apply the appropriate decompression algorithms to the compressed data.

PCDEZIP starts by performing initialisation chores such as installing a control-break vector, saving the starting drive and default directory, scanning the command line arguments, and opening the requested .ZIP file. Then the real work begins.

The first step is to look for compressed files within the .ZIP file. This will be discussed below in the section entitled, 'The ZIP file format'. As it encounters each file, PCDEZIP determines whether the file should be decompressed by checking the filenames supplied on the command line, the switches supplied (/F, /N, and /O), whether the file already exists, the date and time on the existing file, and the answer supplied to any overwrite prompts solicited by PCDEZIP. If PCDEZIP disqualifies the file, it moves on to the next file in the .ZIP file.

If PCDEZIP determines that the file should be decompressed, it then opens the output file. If the /D switch was found on

Viewing a .ZIP file's contents

Processing: DATA.ZIP

Length	Method	Size	Ratio	Date	Time	Name
0	stored	0	0%	09-16-93	11:03	1991\
0	stored	0	0%	09-16-93	11:03	1992\
0	stored	0	0%	09-16-93	11:03	1993\
16826	deflated	3930	77%	09-15-93	15:30	1991\INCOME.DAT
4387	deflated	1770	60%	08-26-93	9:39	1991\INCOME.IDX
18650	deflated	7480	60%	09-14-93	20:23	1992\INCOME.DAT
6949	deflated	2751	61%	09-16-93	10:48	1992\INCOME.IDX
9017	deflated	2197	76%	09-15-93	15:12	1993\INCOME.DAT
2006	deflated	841	58%	09-16-93	10:42	1993\INCOME.IDX
392	deflated	174	56%	09-16-93	11:04	INDEX.DAT
58227		19143	68%			10

PCDEZIP completed normally

Fig 2: Entering PCDEZIP with the /V switch displays the contents of a .ZIP file

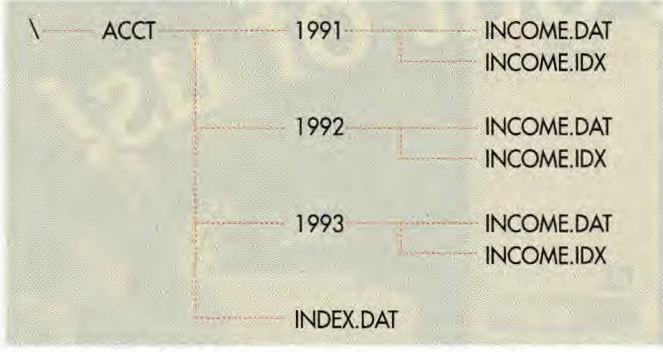


Fig 3: In creating DATA.ZIP, PKZIP used its `-apr` option to capture the subdirectory tree for the `\ACCT` directory

the command line, this process includes building any required directories. Once the output file is open, PCDEZIP decompresses the data and writes the resultant information to the output file.

As decompression progresses, PCDEZIP checks the output file's integrity by using a cyclic redundancy check (CRC). When the .ZIP file is created, the CRC value is calculated using every byte in each of the original, uncompressed files. This value is stored in the .ZIP file. If even a single byte of the decompressed file differs from the original file, the CRC will be different, and PCDEZIP will declare an error in the

contents begin with the letters PK. In fact, if you look through the contents of a .ZIP file carefully, you'll find the letters PK scattered about in several places. That's because these letters — the initials of PKZIP's creator, Phil Katz — form the first two bytes of the signature fields used in the structures that describe the .ZIP file's contents.

Specifically, two structures are used to describe compressed files in a .ZIP file: the local file header and the central directory. The local file header immediately precedes the compressed data for the file it describes. The signature of the local file header is

decompressed file. PCDEZIP then searches for the next file to decompress until it reaches the end of the .ZIP file.

The ZIP file format

If you've ever displayed the contents of a .ZIP file with DOS's TYPE command, you've probably noticed that the file's apparently unintelligible

PK,3,4, where the PK are characters and the 3 and 4 are binary values in the subsequent bytes. (For a fast comparison of the signature value, PCDEZIP reads the 4-byte signature and compares the resulting value against a long value of 0x04034b50. Since Intel microprocessors store values with the least significant byte in the lowest memory address, this value is reversed from the way it would be viewed in a dump of the file.)

The local file header contains several fields. These fields include the filename, compressed file size, uncompressed file size, CRC value, and a value that indicates the compression algorithm used to compress the file. Curiously missing from the local file header is the attribute byte for the file.

The central directory, which comes after all of the compressed file data, is made up of the main file header and an end-of-central-directory record. This file header contains all of the data found in the local file headers together with the file attributes and a pointer to the local file-header structure. The signature for the main file header is PK,1,2, or 0x02014b50.

The third structure found in the .ZIP file is the end-of-central-directory record. Its signature is PK,5,6 (0x06054b50). There is only one of these per .ZIP file, and it immediately follows the last file header. Figure 4 shows the general format of a .ZIP file.

Of these three structures, PCDEZIP uses only the local file header. (This is why PCDEZIP cannot preserve the original file attribute and must decompress all files as read-write.) When PCDEZIP finds a local file header, it uses the information there to determine whether the file should be decompressed and if so, what method to use to decompress the associated file. PCDEZIP recognises the existence of the file header and end-of-central-directory records, but it skips them in search of more local file-header structures.

When you test a .ZIP file using the /T

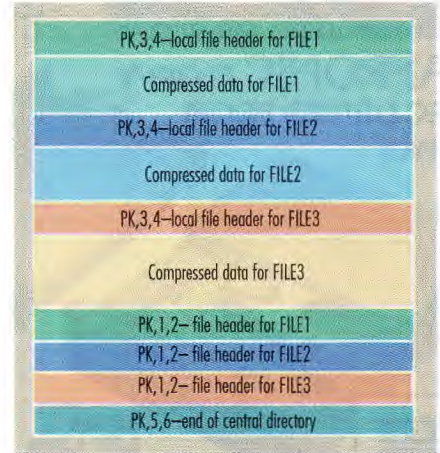


Fig 4: The general layout of a .ZIP file

Quick reference card

PCDEZIP

Purpose: Extracts the contents of .ZIP files downloaded from MagDisk. PCDEZIP replaces the earlier PCUNZIP utility in order to achieve compatibility with Version 2.0 of PKZIP. The new utility is compatible with both versions of PKZIP and provides additional switches and targeting options.

Format: PCDEZIP [switches] zip_file [target\
[filespec [. . .]]

Remarks: The zip_file specified may contain a pathname. By default, PCDEZIP decompresses files into the current directory, but an alternative destination may be specified with the target\
option. Note that when this option is used, the target destination must always contain the trailing backslash shown. The filespec option, which supports wildcards, permits naming specific files to be extracted from the .ZIP file. A help screen, obtained by entering PCDEZIP at the command prompt with no zip_file argument, briefly explains the utility's six switches, which are identified by being preceded with a / or a - character.

The /F switch extracts only files that have an older version (which will be overwritten) in the destination directory. The /N switch extracts these and any files that do not have later-dated versions in the destination directory. The /O switch decompresses all of the files in the .ZIP file and automatically overwrites any same-named files without regard to their date. These three switches disable the normal warning prompt.

The /V switch permits viewing the contents of a .ZIP file without extracting any files, and the /T switch tests the integrity of the zip_file. The /D switch recreates any directory structure stored in the .ZIP file. When no directory information is stored, the /D switch permits the creation of a new destination directory when a target\
argument is also specified.

switch, PCDEZIP performs a full extract and decompress function but does not write the decompressed information to disk. Instead, the purpose of the extraction is to calculate and compare the CRC against that of the original file. If the CRC does not match, PCDEZIP issues a message that the file may be corrupt.

Viewing the contents of the .ZIP file is a significantly different process. When you view a file (using /V), PCDEZIP searches the file for local file headers and prints the contents as they are encountered. After displaying the information, PCDEZIP skips over the compressed data and searches for another header

until the end of file is reached. With PCDEZIP, you can confidently download files compressed with PKZIP, Version 2.0. The new utility even adds to the capabilities of the obsolete PCUNZIP. We hope this makes up for the inconvenience of switching your decompression software.▲