

PKZIP® 2.50  
Command Line  
User's Manual

Copyright © 1997 PKWARE, Inc. All Rights Reserved.

No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any other language in whole or in part, in any form or by any means, whether it be electronic, mechanical, magnetic, optical, manual or otherwise, without prior written consent of PKWARE, Inc.

PKWARE, INC. DISCLAIMS ALL WARRANTIES AS TO THIS SOFTWARE, WHETHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, FUNCTIONALITY, DATA INTEGRITY OR PROTECTION.

PKWARE, the PKWARE logo, and PKZIP are registered trademarks of PKWARE, Inc.

Trademarks of other companies mentioned in this documentation appear for identification purposes only and are the property of their respective companies.

# Table of Contents

<b>Chapter 1</b>	<b>Welcome to PKZIP</b>	<b>1</b>
<hr/>		
<b>Introduction</b>		<b>2</b>
<b>About This Manual</b>		<b>2</b>
<b>Your Work Environment: The Command Line</b>		<b>3</b>
<b>Using Help</b>		<b>3</b>
<b>Getting License Information</b>		<b>4</b>
<b>Getting Version Information</b>		<b>5</b>
<b>Using PKWARE Technical Support</b>		<b>6</b>
Using Internet/Usenet		6
Using CompuServe		7
Using the PKWARE BBS		7
Before You Call		7
Calling a PKWARE Technical Support Representative		8
Using the PKWARE Customer Support Form		8
<b>Chapter 2</b>	<b>The Basics</b>	<b>11</b>
<hr/>		
<b>Introduction</b>		<b>12</b>
<b>Using the Tutorials</b>		<b>12</b>
Conventions Used in the Tutorials		13
Using Sample Files and Directories		13
Preparing Your Workspace		13
<b>.ZIP File Naming Conventions</b>		<b>14</b>
<b>Compressing Files: Learning the Basics</b>		<b>14</b>
Compressing a Single File in the Current Directory		15
Compressing Selected Files in the Current Directory		17
Compressing Files That Match a Pattern		18
Compressing All Files in the Current Directory		19
Compressing Files from a Different Location		20
Specifying a Different Location for the .ZIP File		21
Moving Files into Your .ZIP File		23
Viewing Files Within a .ZIP File		24
Further Information on Compressing Files		26
<b>Extracting Files: Learning the Basics</b>		<b>26</b>
Extracting a Single File From a .ZIP File		27
Extracting Multiple Files From a .ZIP File		29
Overwriting Files that Already Exist in Your Directory		34
Extracting Files to a Specified Directory		34

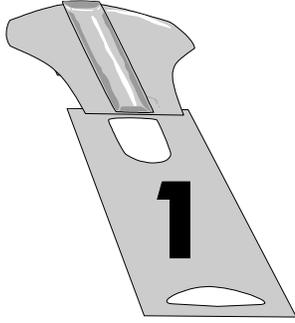
Further Information on Extracting Files	35
<b>Understanding PKZIP Commands and Options</b>	<b>35</b>
Difference between a Command and Option	36
Including an Option in Your Command Line	36
Abbreviating Commands and Options	37
Combining Options	37
Commands and Options That Contain Values	39
Commands and Options: Compression or Extraction?	41
Setting Defaults	41
<b>What's Next</b>	<b>41</b>
<b>Chapter 3    Compressing Files</b>	<b>43</b>
<hr/>	
<b>Introduction</b>	<b>44</b>
Conventions in This Chapter	44
Default Values for Commands and Options	44
<b>Compressing New and Existing Files</b>	<b>45</b>
Compressing All Files in a Directory	45
Compressing New and Modified Files	46
Compressing Only Files That Have Changed	47
Clearing Archive Attributes (DOS, OS/2)	47
<b>Compressing Files in Subdirectories</b>	<b>48</b>
<b>Storing Directory Path Information</b>	<b>49</b>
Additional Methods for Storing Directory Path Information	50
Storing and Recreating Directory Path Information	52
<b>Compressing Files with a LIST File</b>	<b>54</b>
<b>Compressing Files Based on Some Criteria</b>	<b>55</b>
Compressing Files Newer Than a Specified Date	55
Compressing Files Older Than a Specified Date	56
Including Files That Match a Pattern	56
Excluding Files from Being Compressed	58
<b>Compressing Files in a Specific Format</b>	<b>59</b>
<b>Storing File Information</b>	<b>59</b>
Compressing Files That Contain Certain Attributes (DOS, OS/2)	60
Compressing Files Based on File Type (UNIX)	62
Extended Attribute Storage	63
Extended Attributes and 204g Compatibility	65
Removing File Attributes when Compressing (DOS, OS/2)	65
<b>Including Additional Information in a .ZIP File</b>	<b>67</b>
Including a Text Comment	67
Including a Password	69
Including a Header Comment	71

Including a Volume Label (DOS, OS/2)	71
Specifying the Date of a .ZIP File	72
<b>Setting the Compression Level</b>	<b>73</b>
Specifying a Compression Level by Number	73
Specifying a Compression Level by Option	74
<b>Sorting Files Within a .ZIP File</b>	<b>75</b>
<b>Moving Files to a .ZIP File</b>	<b>77</b>
<b>Creating Self-Extracting .ZIP Files</b>	<b>78</b>
Creating a Native Self-Extractor	79
Creating a Regular DOS Self-Extractor	79
Creating a DOS Junior Self-Extractor	80
Creating a Windows 16-Bit Self-Extractor	80
Creating Self-Extractors with the pksfxs.dat file	80
Available Options in the Native Self-Extractor	81
Differences between a Regular DOS and DOS Junior Self-Extractor	81
Converting a .ZIP file to a Self-Extracting file	81
<b>Compressing Files to Diskette (DOS)</b>	<b>82</b>
<b>Chapter 4   Extracting Files</b>	<b>83</b>
<hr/>	
<b>Introduction</b>	<b>84</b>
Conventions in This Chapter	84
Default Values for Commands and Options	84
<b>Extracting New and Existing Files</b>	<b>85</b>
Extracting All Files From a .ZIP File	85
Overriding Default Files to Extract	86
Extracting Newer Versions of Existing Files and New Files	86
Extracting Only Newer Versions of Files	87
<b>Extracting Files Based on Some Criteria</b>	<b>87</b>
Extracting Files Newer Than a Specified Date	88
Extracting Files Older Than a Specified Date	88
Including Files That Match a Pattern	89
Excluding Files from Being Extracted	90
<b>Extracting Files in a Specific Format (OS/2)</b>	<b>90</b>
<b>Determining How to Handle Overwriting Files</b>	<b>91</b>
<b>Retaining Directory Structure while Extracting</b>	<b>92</b>
<b>Sorting Files in the Extract Directory</b>	<b>93</b>
<b>Extracting Files Only for Display</b>	<b>93</b>
<b>Extracting Files with a LIST File</b>	<b>94</b>
<b>Authenticity Verification</b>	<b>95</b>

<b>Chapter 5</b>	<b>Performing Miscellaneous Tasks on .ZIP Files</b>	<b>97</b>
	<b>Introduction</b>	<b>98</b>
	<b>Viewing the Contents of a .ZIP File</b>	<b>98</b>
	Displaying a Brief View of a .ZIP File	99
	Displaying a Detailed View of the .ZIP File	100
	<b>Printing the Contents of a .ZIP File (DOS, OS/2)</b>	<b>101</b>
	<b>Testing the Integrity of a .ZIP File</b>	<b>101</b>
	<b>Previewing Command and Option Operations</b>	<b>102</b>
	<b>Fixing a Corrupt .ZIP File</b>	<b>103</b>
	<b>Create a Temporary .ZIP File on a Alternate Drive</b>	<b>104</b>
	<b>Suppressing Screen Output</b>	<b>105</b>
<b>Chapter 6</b>	<b>Changing Defaults Using the Configuration File</b>	<b>107</b>
	<b>Introduction</b>	<b>108</b>
	<b>Viewing the Configuration File</b>	<b>108</b>
	<b>Changing a Default Value</b>	<b>110</b>
	<b>Resetting Original Defaults</b>	<b>111</b>
	Resetting All Defaults	112
	Resetting Individual Defaults	112
<b>Chapter 7</b>	<b>Command Characteristics</b>	<b>113</b>
	<b>Introduction</b>	<b>114</b>
	<b>Changing Date and Time Environment Variables</b>	<b>114</b>
	<b>Changing the LIST Character for LIST Files</b>	<b>116</b>
	<b>Changing the Command/Option Character</b>	<b>117</b>

<u>Appendix A</u>	<u>PKZIP Options: A Quick Reference</u>	<u>119</u>
<u>Appendix B</u>	<u>Error and Warning Messages</u>	<u>143</u>
<u>Appendix C</u>	<u>Frequently Asked Questions</u>	<u>153</u>
<u>Appendix D</u>	<u>How Does PKZIP Work</u>	<u>161</u>
<u>Index</u>		<u>171</u>





# Welcome to PKZIP

## Overview:

This chapter contains an introduction to PKZIP, an explanation of the contents of this manual, and information about the technical support services offered to you as a valued PKZIP user.

**Note:** It is highly recommended that you review the **Getting Started Manual** before proceeding.

## Sections in This Chapter:

- Introduction
- About This Manual
- Your Work Environment: The Command Line
- Using Help
- Getting License Information
- Getting Version Information
- Using PKWARE Technical Support

## Introduction

---

Compressing files into a .ZIP archive is quick and easy with PKZIP. The powerful but simple to use PKZIP command line gives you the ability to fully customize your data compression tasks. Before you create a .ZIP file, consider the following:

- Do you prefer speed or compression?
- Do you want directory path information stored in your .ZIP file?
- Will you send .ZIP files to others? If so, would a self-extracting .ZIP benefit them?
- Do you want to protect your .ZIP file with a password?

With PKZIP, you can create two types of .ZIP files: regular and self-extracting. A regular .ZIP file contains only the compressed files. (i.e. you need PKZIP installed on your system to extract the files) A self-extracting .ZIP file contains the compressed files "and" built-in instructions that allow you to extract files without using PKZIP (it also contains a different extension, .exe). This is helpful when you send .ZIP files to those who don't have access to PKZIP or other .ZIP compatible programs.

## About This Manual

---

This manual is organized for the novice and expert user alike. If you are a new PKZIP user, you will find the tutorials in this manual (Chapter 2) helpful. If you have previously used other versions of PKZIP (e.g. PKZIP for DOS 2.04), you may find that although the base functionality has been preserved, the command line implementation is different with new commands and options. If you are an advanced user, *Appendix A: PKZIP Options: A Quick Reference* on page 119 might be all the reference material you need.

## Your Work Environment: The Command Line

---

PKZIP is a command-line based product, which means your work area is a character-based command line. Using PKZIP consists primarily of typing a command and pressing ENTER. Command prompt conventions allow you to extensively customize your compression and extraction tasks.

**Note:** Although this manual isn't an operating system manual, it does contain general reminders on conventions as they relate to a task or tutorial.

## Using Help

---

In addition to the information provided in this manual, PKZIP also contains on-line help for your convenience. The on-line help contains information on every PKZIP command, option, and sub-option.

To access on-line help:

1. At the command prompt, type the following and press ENTER:

```
pkzip25 -help
```

A screen with PKZIP version and usage information will appear as well as the following prompt:

```
Press any key to continue or <Esc> to exit
```

Press any key. A screen with all available PKZIP commands and options will appear as well as the following prompt:

```
Enter the command or option for additional help, <Esc> to exit  
Command/Option?
```

Enter a command or option at the prompt and press ENTER. The help information for that particular command or option will appear.

2. To bypass the command/option menu and go directly to a help file for a particular command or option, type the **help** command followed by an equal sign (=) and the command or option for which you need on-line assistance. For example, to access on-line help for the **add** command, type the following at the command prompt and press ENTER:

```
pkzip25 -help=add
```

The help information for the **add** command will appear.

## Getting License Information

---

To display the PKZIP license information on your screen, do the following:

1. At the command prompt, type the following and press ENTER:

```
pkzip25 -license
```

If you have the registered version of PKZIP, information similar to the following appears:

```
PKZIP(R) Version 2.50 FAST! Compression Utility
Copyright 1989-1997 PKWARE Inc. All Rights Reserved. Registered Version
PKZIP Reg. U.S. Pat. and Tm. Off. Patent No. 5,051,745
```

```
PKWARE DISCLAIMS ALL WARRANTIES AS TO THIS SOFTWARE, WHETHER EXPRESS OR
IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANT-
ABILITY, FITNESS FOR A PARTICULAR PURPOSE, FUNCTIONALITY OR DATA INTEGRITY
OR PROTECTION. See documentation for further disclaimers and information.
```

```
For use on multiple machines or to use to distribute your information
or software, a separate license is required. Please see the LICENSE.TXT file
for further information.
```

```
This is a registered version and is only for use on those
machine(s) for which it is licensed.
```

```
This version is NOT TO BE DISTRIBUTED as Shareware.
```

```
PKWARE, Inc.
9025 N. Deerwood Drive
Brown Deer, WI 53223
```

## Getting Version Information

---

To determine the version of PKZIP you are using, do the following:

1. At the command prompt, type the following and press ENTER:

```
pkzip25 -version
```

Information similar to the following appears:

```
PKZIP(R) Version 2.50 FAST! Compression Utility
Copyright 1989-1997 PKWARE Inc. All Rights Reserved. Registered Version
PKZIP Reg. U.S. Pat. and Tm. Off. Patent No. 5,051,745
Version 2.50
```

Advanced users of PKZIP often require more detailed version information to assist in automating compression and extraction operations. PKZIP has the capability to return the version number as a value to the shell. The version number will be returned as a positive integer value less than 256. This value is only returned to the shell and is therefore not viewable by the user. This functionality can be useful to verify PKZIP version numbers in the context of a .BAT file or shell script. The default action is to return the major number of the release. For example, for version 2.5 the return value will be 2. Using sub-options with the **version** option will return precise values of the release number to the shell. The available sub-options and the values they return are outlined in the following table:

Sub-Option:	PKZIP Returns:	For example:
<b>major</b>	The major release number. For example, if the version number is 2.5, the value returned is <2>.	pkzip25 -version=major
<b>minor</b>	The minor number of the release. For example, if the version number is 2.5, the value returned is <5>.	pkzip25 -version=minor
<b>step</b>	The step or patch value. For example, if the version is 2.04.01, the step value returned is <1>.	pkzip25 -version=step

## Using PKWARE Technical Support

---

We hope that using PKZIP is an enjoyable and trouble-free experience. Our goal is to provide you with the best product and documentation possible. However, problems and questions might arise. In most cases, you'll be able to answer those questions or solve the problems by referring to the **Getting Started Manual**, the on-line help or the manual you are now reading. If however you need further assistance, you can reach the PKWARE Technical Support Department in the following ways:

- Internet/Usenet.
- CompuServe.
- PKWARE Bulletin Board Service (BBS).
- Faxing/Calling a Technical Support Representative.

Remember to refer to this manual **before** you use one of these options.

### Using Internet/Usenet

PKWARE products are frequently discussed in the newsgroup comp.compression. Internet newsgroups are not supported directly by PKWARE. You will find a customer assistance area on PKWARE's World Wide Web site. Additionally, you may use the Internet/Usenet to electronically mail PKWARE Technical Support. PKWARE's Internet addresses are as follows:

- World Wide Web: <http://www.pkware.com>
- Electronic Mail: [support@pkware.com](mailto:support@pkware.com)

## Using CompuServe

PKWARE technical support representatives maintain a discussion forum on CompuServe. To reach the forum, type GO PKWARE from any CompuServe prompt. PKWARE is located in area 11. PKWARE's CompuServe ID is 75300,730

## Using the PKWARE BBS

PKWARE maintains its own BBS. It is normally available 24 hours a day and supports baud rates from 1200 bps to 33.6 kbps.

To use this BBS, set your communications software to the following:

8 bits; 1 stop bit; No parity

For your convenience, the PKWARE BBS has multiple telephone lines, offers the latest in PKWARE shareware, and gives you the ability to download many other files.

You can reach the PKWARE BBS at the following number:

- (414) 354-8670

## Before You Call

The more information you provide, the easier it will be for PKWARE Technical Support Representatives to help you. Before you use one of the technical support services, please have the following information handy:

- The version number of PKZIP (refer to *Getting Version Information*).
- The "exact" commands you typed into your computer prior to noticing the problem and any error messages that appear.

## **Calling a PKWARE Technical Support Representative**

The most direct way to receive assistance about PKZIP is to call a PKWARE Technical Support Representative on the telephone. The PKWARE Technical Support Department is available Monday through Friday from 9:00a.m. to 5:00p.m. Central Standard Time.

Before you call, we recommend that you FAX the necessary configuration information to PKWARE. To help you organize this information, use the Customer Support Form that follows. This gives the technical support representative some time to review your problem and offer a solution. Please indicate on the FAX that you will be calling.

The PKWARE FAX number is:

- (414) 354-8559

If you have determined that your problem is serious and needs immediate attention, call the PKWARE Technical Support Department at the following number:

- (414) 354-8699

## **Using the PKWARE Customer Support Form**

For your convenience, a Customer Support Form has been provided to help you record the information you need to accurately report the problem you are experiencing. We recommend that you make several copies of this form.

REMEMBER! Include as much information on this form as possible.

# Customer Support Form

## Customer Information

Your Name:	
Company Name:	
Voice Phone Number:	
FAX Phone Number:	
Address:	
City and State:	
ZIP code and Country:	
Email Address:	

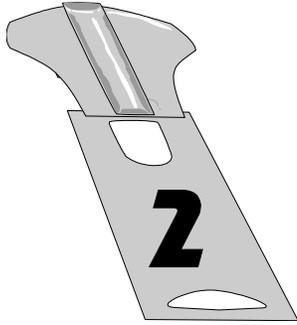
## System Configuration Information

Software Version:	
Computer Model/Make:	
Operating System Version:	
Drive Model/Vendor:	

## Information

Problem:	
Description: (Use Next Page)	

**Notes:**



## The Basics

### Overview:

This chapter contains information to get you started with PKZIP, including quick, easy-to-follow instructions on performing basic compression (PKZIP) and extraction (PKUNZIP) tasks. Tutorials help you practice, by systematically guiding you through each task.

### Sections in This Chapter:

- Introduction
- Using the Tutorials
- Compressing Files: Learning the Basics
- Extracting Files: Learning the Basics
- Understanding PKZIP Commands and Options
- What's Next?

## Introduction

---

This chapter introduces you to basic compression and extraction tasks using PKZIP. This includes using command-line conventions to help customize your data compression and extraction tasks. For example, you may specify a destination location for your .ZIP file even if you are not compressing files in that particular location. You may also use conventions such as "\*" to represent "all" files. PKZIP incorporates these conventions to help make using PKZIP easy as well as flexible.

In addition to using conventions, PKZIP contains its own set of special commands and options that even further customize compression and extraction tasks. These commands and options are detailed in subsequent chapters as well as the chapter you are now reading. Specifically, this chapter contains tutorials that will allow you to practice with PKZIP on files that were created and copied to a special "tutorial" directory. Refer to the *Creating the Tutorial Directory and Files* section, in the **Getting Started Manual** for specific information on setting up your tutorial workspace.

## Using the Tutorials

---

To help you learn PKZIP, this chapter contains several tutorials. These tutorials progress from simple to complex features. Each tutorial is preceded by a brief explanation of the topic(s) that the tutorial represents. For each tutorial, you will be asked to perform a specific PKZIP task. The results of "what you type" are reviewed every step of the way.

## Conventions Used in the Tutorials

This manual was written for several operating system and platform versions of the command line implementation of PKZIP. To avoid confusion, we have tried to minimize references to specific operating systems or platforms. There may be instances in the tutorials where it is necessary to refer to an operating system or platform convention. Since PKZIP's command line interface is virtually identical between the various operating system/platform versions, the examples used in this manual are, for the most part, universal. However, keep in mind that what you see on your screen might differ slightly from what you see in the examples.

Typically, in command line examples, we only reference what you would actually type at the command prompt, as in the following command line example:

```
pkzip25 -add test.zip text.doc
```

## Using Sample Files and Directories

The tutorials in this manual take you through compressing and extracting "specific" files from a "specific" tutorial directory. Feel free to use your own files if you choose to practice on them. However, keep in mind that the results that are displayed are specific to the files that these tutorials use as samples.

**Note:** These tutorials assume that you have installed PKZIP in the default installation directory. Additionally, the tutorials assume that you can access PKZIP from any directory without specifying a file path. Refer to the **Getting Started Manual** for information on how to properly install and configure PKZIP.

## Preparing Your Workspace

To follow the tutorials in this manual step-by-step, you must create a "working" directory. Consider it a "temporary" directory to be used only for the tutorials. This helps to ensure that your permanent directories and files are not deleted or damaged while you're practicing with PKZIP. Refer to the *Creating the Tutorial Directory and Files* section in the **Getting Started Manual** for specific information on setting up your tutorial workspace.

## **.ZIP File Naming Conventions**

---

The commands and options discussed in the following sections and chapters of this manual are entered (except for a few standalone commands) on a command line along with the name of a .ZIP file. If you enter the file name as "file.zip" or "file.exe", PKZIP will *not* modify the file name. However, any file name entered without a .zip or .exe extension will have an extension added by PKZIP unless the file name ends with a "trailing dot" (e.g. "file."). Specifying a "trailing dot" with your file name on the command line tells PKZIP *not* to append an extension to the specified file name. For example, specifying "t." as the archive name would result in a file called "t". Conversely, specifying "t" as the archive name would result in a file called "t.zip". Other "dots" in a file name are ignored and left unchanged and only the trailing dot, if present, is used to suppress the adding of an extension to the .ZIP file name. The *nozipextension* option does the same thing.

Systems that do not support more than one "dot" per file name will also suppress the extension if any dot is present in the file name. For example, on OS/2 FAT, entering "file.wow" will result in a file called "file.wow", with no .ZIP extension added by PKZIP, even though the dot is not at the end of the file name. On the other hand, entering "file.wow" on Win32 FAT, UNIX, or OS/2 (H.P.F.S. & N.T.F.S.) will result in a file called "file.wow.zip".

## **Compressing Files: Learning the Basics**

---

This section contains information on how to perform simple data compression. You perform all PKZIP tasks from a command prompt.

The following are the minimum elements you need to type to compress files:

- The word **pkzip25**, which is the main PKZIP command.
- A dash (-) followed by the word **add** (which is the base command used to compress files).

- The name of the .ZIP file.
- The names of the file(s) you want to compress.

For example, to compress a file called test.txt into a .ZIP file called temp.zip, type the following:

```
pkzip25 -add temp.zip test.txt
```

When you compress files, the "pkzip25" and "-add" are constant. Every other element of the command varies, depending on the files being compressed in addition to their location. The sections that follow cover a wide range of compression (and extraction) possibilities in addition to tutorials to guide you.

**Note:** Many of the following tutorials instruct you to change from one directory to another. This operation is carried out with the help of the **cd** command. The **cd** command changes the current directory to a directory that you specify on the command line. Consult your operating system documentation for more information on the **cd** command.

## Compressing a Single File in the Current Directory

The simplest PKZIP task is compressing a single file from the current working directory, and storing the resulting .ZIP file in that same directory. In this scenario, you are not retrieving files to compress from other directories, nor are you placing the .ZIP file in a different directory.

### Tutorial A

Compress a single file (red.txt) into a .ZIP file in your current working directory.

Follow these steps:

1. From the command prompt, change to the PKZIP Tutorial (tut) directory that you have created. For information on creating the PKZIP Tutorial directory, refer to the **Getting Started Manual**.

2. To compress the file red.txt into the .ZIP file called test.zip, type the following command and press ENTER:

```
pkzip25 -add test.zip red.txt
```

Your file starts to compress, and your screen looks like the following:

```
PKZIP(R) Version 2.50 FAST! Compression Utility
Copyright 1989-1997 PKWARE Inc. All Rights Reserved. Registered Version
PKZIP Reg. U.S. Pat. and Tm. Off. Patent No. 5,051,745
```

```
Creating .ZIP: test.zip
Adding File: red.txt      Deflating   (62.1%), done.
```

The first three lines contain the PKZIP banner information (name of the product, date, version and so on).

The last two lines list the name of the .ZIP file you created, the file(s) that were compressed, and the percentage that the file was compressed.

You have now successfully compressed a file into a .ZIP file. Before you continue with the next section, let's take a look at your tut directory.

3. View the contents of the tut directory by typing one of the following commands:

➤ From a UNIX based command prompt, type:

```
ls -l
```

A file listing similar to the following will appear:

```
-rw-rw-rw-  1 user  pkware  1030 June 30 2:50 black.tut
-rw-rw-rw-  1 user  pkware  4185 June 30 2:50 blue.fil
-rw-rw-rw-  1 user  pkware  5920 June 30 2:50 brown.doc
-rw-rw-rw-  1 user  pkware  1030 June 30 2:50 gold.tut
-rw-rw-rw-  1 user  pkware   591 June 30 2:50 green.doc
-rw-rw-rw-  1 user  pkware   591 June 30 2:50 orange.fil
-rw-rw-rw-  1 user  pkware  1030 June 30 2:50 pink.tut
-rw-rw-rw-  1 user  pkware   591 June 30 2:50 purple.txt
-rw-rw-rw-  1 user  pkware  4185 June 30 2:50 red.txt
-rw-rw-rw-  1 user  pkware  4185 June 30 2:50 tan.txt
-rw-r--r--  1 user  pkware  1702 June 30 2:59 test.zip
-rw-rw-rw-  1 user  pkware  5920 June 30 2:50 white.doc
-rw-rw-rw-  1 user  pkware  5920 June 30 2:50 yellow.doc
```

➤ From a DOS based command prompt, type:

```
dir
```

A file listing similar to the following will appear:

```
.          <DIR>      06-30-97 12:00a .
..         <DIR>      06-30-97 12:00a ..
ORANGE    FIL        591 06-30-97 2:50a orange.fil
YELLOW    DOC       5,920 06-30-97 2:50a yellow.doc
RED       TXT       4,185 06-30-97 2:50a red.txt
GREEN     DOC        591 06-30-97 2:50a green.doc
PURPLE    TXT        591 06-30-97 2:50a purple.txt
WHITE     DOC       5,920 06-30-97 2:50a white.doc
BROWN     DOC       5,920 06-30-97 2:50a brown.doc
PINK      TUT        1,030 06-30-97 2:50a pink.tut
TEST      ZIP        1,702 06-30-97 2:59a test.zip
TAN       TXT       4,185 06-30-97 2:50a tan.txt
BLACK     TUT        1,030 06-30-97 2:50a black.tut
BLUE     FIL       4,185 06-30-97 2:50a blue.fil
GOLD     TUT        1,030 06-30-97 2:50a gold.tut
13 file(s)      36,880 bytes
2 dir(s)        87,945,216 bytes free
```

Note the test.zip file you have created. The file you compressed into the .ZIP file still appears in the current directory. That's because when you compress files, they remain in their original location, as well as inside the .ZIP file, compressed.

You can choose to remove or "move" the files into the .ZIP file so that they exist only in the .ZIP file after compression. The *Moving Files Into Your .ZIP File* section on page 23 shows you how to do this.

## Compressing Selected Files in the Current Directory

With PKZIP, you can compress specific selected files into your .ZIP file. To do this, you simply type the files in your command, including a space between each one.

Tutorial B

Compress the following files into the test.zip file:

- green.doc
- blue.fil

Follow these steps:

1. From the command prompt, type the following and press ENTER:

```
pkzip25 -add test.zip green.doc blue.fil
```

Your file starts to compress, and output similar to following appears:

```
PKZIP(R) Version 2.50 FAST! Compression Utility
Copyright 1989-1997 PKWARE Inc. All Rights Reserved. Registered Version
PKZIP Reg. U.S. Pat. and Tm. Off. Patent No. 5,051,745

Updating .ZIP: test.zip
Adding File: green.doc Deflating (39.3%), done.
Adding File: blue.fil Deflating (62.1%), done.
```

This particular example added the specified files to the existing test.zip file. As you can see in the screen output above, PKZIP reports that it is "Updating .ZIP: test.zip" The .ZIP archive now contains any file(s) we added in the previous tutorial as well as the file(s) we added in this tutorial.

## Compressing Files That Match a Pattern

PKZIP allows you to compress "only" files of a specific file pattern or extension. For example, you might want to compress only .doc files or .bmp files. On the other hand, you might want to compress all files that begin with the letter "p" or any other character match.

Using PKZIP conventions for typing file patterns (commonly called "wildcards"), follow these guidelines:

- For files of a specific extension, type an asterisk, a period, and the extension. For example, for ASCII text files (.txt extension), you would type **\*.txt**
- For files that begin with a specific character or characters, type the character(s), then an asterisk (\*). For example, for all files that begin with res, you would type **res\***
- For files that end with a specific character pattern, type an asterisk (\*), then the character(s). For example, for all files that end in "all" you would type **\*all**

**Note:** For other "wildcard" conventions and general command-line conventions, refer to your Operating System documentation.

## Tutorial C

Compress all files in your tut directory that end with \*.doc

Follow these steps:

1. From your command prompt, type the following and press ENTER:

```
pkzip25 -add test.zip *.doc
```

The following appears on your screen:

```
PKZIP(R) Version 2.50 FAST! Compression Utility  
Copyright 1989-1997 PKWARE Inc. All Rights Reserved. Registered Version  
PKZIP Reg. U.S. Pat. and Tm. Off. Patent No. 5,051,745
```

```
Updating .ZIP: test.zip  
Updating File: green.doc Deflating (39.3%), done.  
Adding File: yellow.doc Deflating (59.4%), done.  
Adding File: white.doc Deflating (59.4%), done.  
Adding File: brown.doc Deflating (59.4%), done.
```

When PKZIP is finished, the command prompt reappears.

## Compressing All Files in the Current Directory

PKZIP allows you to compress all files in a specified directory. In the previous tutorials you compressed files by specifying either a specific file name (e.g. red.txt) or file pattern (e.g. \*.doc). When you compress "all" files, you only have to type the name of the .ZIP file with the **add** command. PKZIP will assume that you wish to compress all files in the current directory. If, however, you wish to specify "all" files in another directory, you must use the convention for specifying "all" files (e.g. "\*\*").

## Tutorial D

Compress all the files in your tut directory into the test.zip file.

Follow these steps:

1. From your command prompt, type the following and press ENTER:

```
pkzip25 -add test.zip
```

Your files start to compress. Because you are compressing all of the files in your tut directory, this will take a little longer.

Output similar to the following appears:

```
PKZIP(R) Version 2.50 FAST! Compression Utility
Copyright 1989-1997 PKWARE Inc. All Rights Reserved. Registered Version
PKZIP Reg. U.S. Pat. and Tm. Off. Patent No. 5,051,745

Updating .ZIP: test.zip
Updating File: red.txt      Deflating (62.1%), done.
Updating File: green.doc   Deflating (39.3%), done.
Updating File: blue.fil    Deflating (62.1%), done.
Updating File: yellow.doc  Deflating (59.4%), done.
Updating File: white.doc   Deflating (59.4%), done.
Updating File: brown.doc   Deflating (59.4%), done.
Adding File: tan.txt       Deflating (62.1%), done.
Adding File: orange.fil    Deflating (39.3%), done.
Adding File: black.tut     Deflating (61.0%), done.
Adding File: purple.txt    Deflating (39.3%), done.
Adding File: pink.tut      Deflating (61.0%), done.
Adding File: gold.tut      Deflating (61.0%), done.
```

The phrase *Updating File* appears in front of some files, while *Adding File* appears in front of others. This is because in this tutorial you compressed files that were already compressed in the previous tutorials. The phrase *Updating File* appears with those particular files.

## Compressing Files from a Different Location

When you compress files, you don't have to be in the directory where those files reside. You can compress them from anywhere on your computer. Instead of just typing the file names, you type the directory path along with the file names.

## Tutorial E

In this tutorial, you will switch locations to the directory that is directly above (parent to) the PKZIP 2.5 Command Line install directory and compress files that appear in the PKZIP tutorial directory.

Follow these steps:

1. Go to the directory that is directly above the PKZIP install directory. (e.g. c:\pkware or /usr/local/bin/pkware)
2. To compress the file called purple.txt from the PKZIP tutorial directory into a file called test123.zip file and place the test123.zip file into the current directory, type one of the following command lines and press ENTER:

➤ From a UNIX based command prompt, type:

```
pkzip25 -add test123.zip pkzip25/tut/purple.txt
```

➤ From a DOS based command prompt, type:

```
pkzip25 -add test123.zip pkzip25\tut\purple.txt
```

Your files start to compress, and the following appears:

```
PKZIP(R) Version 2.50 FAST! Compression Utility  
Copyright 1989-1997 PKWARE Inc. All Rights Reserved. Registered Version  
PKZIP Reg. U.S. Pat. and Tm. Off. Patent No. 5,051,745
```

```
Creating .ZIP: test123.zip  
Adding File: purple.txt Deflating (39.3%), done.
```

The test123.zip file is created in the current directory as that is the directory in which you typed the command. You can also specify a different location in which to create or update a .ZIP file. The next section shows you how to do that.

## Specifying a Different Location for the .ZIP File

Until now, you have created a .ZIP file in the current directory. With PKZIP, you can specify a "different" location for the .ZIP file right in the command line. Simply include the location, or directory path, in your command.

## Tutorial F

While remaining in the directory that is parent to the PKZIP install directory, compress the file called gold.tut, which is in PKZIP Tutorial directory, and place it in the test.zip file, which exists in PKZIP Tutorial directory as well.

Follow these steps:

1. Verify that you are in the appropriate directory.
2. To compress gold.tut into the test.zip file, and place the updated .ZIP file in the PKZIP Tutorial directory, type one of the following command lines and press ENTER:

➤ From a UNIX based command prompt, type:

```
pkzip25 -add pkzip25\tut\test.zip pkzip25\tut\gold.tut
```

➤ From a DOS based command prompt, type:

```
pkzip25 -add pkzip25\tut\test.zip pkzip25\tut\gold.tut
```

Your files start to compress, and a screen similar to the following appears:

```
PKZIP(R) Version 2.50 FAST! Compression Utility  
Copyright 1989-1997 PKWARE Inc. All Rights Reserved. Registered Version  
PKZIP Reg. U.S. Pat. and Tm. Off. Patent No. 5,051,745
```

```
Updating .ZIP: pkzip25\tut\test.zip  
Updating File: gold.tut Deflating (61.0%), done.
```

## Moving Files into Your .ZIP File

Normally, the original files that you compress remain on your hard drive after you compress them. That is the default action when you use the **add** command. PKZIP allows you to *remove* the files from your hard drive location *after* they are compressed into the .ZIP file.

**CAUTION:** Be careful when "moving" files from your hard drive to a .ZIP archive. If the .ZIP archive is subsequently damaged, lost, or deleted, any files contained therein will be damaged, lost, or deleted with it. Regularly back up your .ZIP archives to avoid such problems.

To remove files after they are compressed, use the **move** option along with the **add** command. To include an option in your PKZIP command, you simply type that option - preceded by a dash - after the **add** command. The following tutorial shows you how to do this.

**Note:** For basic information on PKZIP options, refer to the *Understanding PKZIP Commands and Options* section on page 35.

### Tutorial G

In this tutorial, you are going to compress the file called pink.tut into the test.zip file, and "remove" pink.tut from your tut directory.

Follow these steps:

1. Change back to the PKZIP Tutorial directory.
2. Type the following and press ENTER:

```
pkzip25 -add -move test.zip pink.tut
```

Your file starts to compress, and your screen looks similar to the following:

```
PKZIP(R) Version 2.50 FAST! Compression Utility
Copyright 1989-1997 PKWARE Inc. All Rights Reserved. Registered Version
PKZIP Reg. U.S. Pat. and Tm. Off. Patent No. 5,051,745

Updating .ZIP: test.zip
Updating File: pink.tut      Deflating    (61.0%), done.

Moving 1 files...
Moved File: pink.tut      , done.
```

Note that PKZIP performed two tasks: Updating and Moving. After compressing the file, PKZIP removed it from the tut directory.

3. To verify that the file you compressed was removed from your directory, type one of the following commands and press ENTER:

➤ From a UNIX based command prompt, type:

```
ls -l
```

➤ From a DOS based command prompt, type:

```
dir
```

Verify from the listing that the pink.tut file no longer exists as an individual file in the PKZIP Tutorial directory. The pink.tut file does however exist, compressed, in the test.zip file.

## Viewing Files Within a .ZIP File

In the previous section, you learned how to move a file from the hard drive into a .ZIP archive. The file you moved still exists in your test.zip file. You can use the **view** command to verify that this file was indeed archived in the .ZIP file. The **view** command allows you to list the files that exist within a .ZIP file. Because you are not compressing files - only viewing them - you do not have to include the **add** command.

## Tutorial H

View the files within the test.zip file.

Follow these steps:

1. Verify that you are in the PKZIP tutorial directory.
2. To view the files within the test.zip file, type the following and press ENTER:

```
pkzip25 -view test.zip
```

PKZIP displays a file listing of all of the files contained in the test.zip file. The listing will look similar to the following:

```
PKZIP(R) Version 2.50 FAST! Compression Utility
Copyright 1989-1997 PKWARE Inc. All Rights Reserved. Registered Version
PKZIP Reg. U.S. Pat. and Tm. Off. Patent No. 5,051,745

Viewing .ZIP: test.zip

Length  Method  Size  Ratio   Date      Time      CRC-32  Attr  Name
-----  -
4185   DeflatN  1590  62.1%  07/30/1997  2:50a  208fe014  --w-  tan.txt
4185   DeflatN  1590  62.1%  07/30/1997  2:50a  208fe014  --w-  red.txt
4185   DeflatN  1590  62.1%  07/30/1997  2:50a  208fe014  --w-  blue.fil
1030   DeflatN  402   61.0%  07/30/1997  2:50a  70856c85  --w-  pink.tut
1030   DeflatN  402   61.0%  07/30/1997  2:50a  70856c85  --w-  black.tut
1030   DeflatN  402   61.0%  07/30/1997  2:50a  70856c85  --w-  gold.tut
5920   DeflatN  2409  59.4%  07/30/1997  2:50a  8687cc4f  --w-  brown.doc
5920   DeflatN  2409  59.4%  07/30/1997  2:50a  8687cc4f  --w-  yellow.doc
5920   DeflatN  2409  59.4%  07/30/1997  2:50a  8687cc4f  --w-  white.doc
591    DeflatN  359   39.3%  07/30/1997  2:50a  d00756d3  --w-  green.doc
591    DeflatN  359   39.3%  07/30/1997  2:50a  d00756d3  --w-  orange.fil
591    DeflatN  359   39.3%  07/30/1997  2:50a  d00756d3  --w-  purple.txt
-----  -
35178                14280  59.5%                                12
```

For more information on the **view** command, see the *Viewing the Contents of a .ZIP File* section on page 98.

**Note:** The above **view** list was generated from a DOS command line. In a UNIX **view** listing, the "Attr" column would be replaced by a "Mode" column with permission numbers for each respective file.

## Further Information on Compressing Files

The tutorials in the previous sections presented a sample of the file compression options available to you. For example, you can compress "only" files that are "newer" than the files already stored in the .ZIP file. You can also set the compression level, prioritizing either speed or compression.

For information on these and other compression features, refer to *Chapter 3 - Compressing Files* on page 43.

## Extracting Files: Learning the Basics

---

With PKZIP, you can "extract" files just as easily as you can compress them. This section shows you how to "extract" files back to their original size. You can extract all files from a .ZIP file, or just one file. You can also extract only the files that are newer than the files with the same name on your hard drive.

When you extract a file, you have to include at least the following in your command, including a space between each element:

- The word ***pkzip25***, which is the main PKZIP command.
- A dash (-) followed by the word ***extract*** (which is the base command used to extract files).
- The name of the .ZIP file.

Including these three elements extracts "all" the files contained in the .ZIP file into the current directory. To extract only specific files from the .ZIP file, you would include the following:

- The name of the file(s) you want to extract, or a file pattern.

For example, to extract a file called `cover.txt` from a compressed file called `temp.zip` into the current directory, you would type the following and press ENTER:

```
pkzip25 -extract temp.zip cover.txt
```

The tutorials that follow cover several "extraction" scenarios. For complete information on extracting files, refer to *Chapter 4 - Extracting Files* on page 83.

## Extracting a Single File From a .ZIP File

The simplest task for extracting files is to extract a single file into the current directory. To make it easy, we are going to create an "extract" directory under your `tut` directory. We'll call this directory `ext`. You will also copy the `test.zip` file into this directory.

Before you extract files, create a directory called `ext` and copy the `test.zip` file into that directory.

Follow these steps:

1. Verify that you are in the PKZIP Tutorial directory.
2. To create the `ext` directory, type the following and press ENTER:

```
mkdir ext
```

3. To copy the `test.zip` file into the `ext` directory, type the following and press ENTER:

➤ From a UNIX based command prompt, type:

```
cp test.zip ext
```

➤ From a DOS based command prompt, type:

```
copy test.zip ext
```

4. Now change to the ext directory. To do so, type the following and press ENTER:

```
cd ext
```

Tutorial I

Extract the file called red.txt from the test.zip file.

Follow these steps:

1. Verify that you are in the ext directory.
2. Type the following and press ENTER:

```
pkzip25 -extract test.zip red.txt
```

PKZIP extracts the file. Your screen will look similar to the following:

```
PKZIP(R) Version 2.50 FAST! Compression Utility  
Copyright 1989-1997 PKWARE Inc. All Rights Reserved. Registered Version  
PKZIP Reg. U.S. Pat. and Tm. Off. Patent No. 5,051,745
```

```
Extracting files from .ZIP: test.zip  
Inflating: red.txt
```

The first three lines contain the PKZIP banner information (name of the product, date, version and so on).

The fourth line contains the task PKZIP performed (in this case, *Extracting*) followed by the name of the .ZIP file (in this case, test.zip).

The fifth line contains the word *Inflating*, then the name of the file being extracted (in this case, red.txt). *Inflating* is an internal method of extraction that is meaningful only to the program.

3. To confirm that red.txt has been extracted into the ext directory, type one of the following commands and press ENTER:

➤ From a UNIX based command prompt, type:

```
ls -l red.txt
```

A listing similar to the following appears:

```
-rw-rw-rw- 1 user pkware 4185 June 30 2:50 red.txt
```

➤ From a DOS based command prompt, type:

```
dir red.txt
```

A listing similar to the following appears:

```
RED      TXT      4,185 06-30-97 2:50a red.txt
1 file(s) 4,185 bytes
0 dir(s) 88,793,088 bytes free
```

Note that red.txt has been extracted into the ext directory.

## Extracting Multiple Files From a .ZIP File

PKZIP allows you to extract multiple files from a .ZIP file at one time. This section shows you how to extract:

- Selected individual files.
- Files that match a specific file pattern. For example, files ending with a .txt extension.
- All files in the directory.

To prepare your ext directory to extract multiple files, first remove the red.txt file that you extracted in the previous tutorial.

Follow these steps:

1. Verify that you are in the ext directory.
2. Type one of the following commands and press ENTER:

➤ From a UNIX based command prompt, type:

```
rm red.txt
```

➤ From a DOS based command prompt, type:

```
del red.txt
```

The red.txt file is deleted. The only file that should be in the ext directory is test.zip.

### **Extracting Selected Files**

With PKZIP, you can extract specific selected files from your .ZIP file. To do this, you simply type the files in your command, including a space between each one.

Tutorial J

Extract the following files from the test.zip file:

- green.doc
- blue.fil

Follow these steps:

1. Verify that you are in the ext directory.
2. Type the following and press ENTER:

```
pkzip25 -extract test.zip green.doc blue.fil
```

A screen similar to the following appears:

```
PKZIP(R) Version 2.50 FAST! Compression Utility
Copyright 1989-1997 PKWARE Inc. All Rights Reserved. Registered Version
PKZIP Reg. U.S. Pat. and Tm. Off. Patent No. 5,051,745
```

```
Extracting files from .ZIP: test.zip
Inflating: blue.fil
Inflating: green.doc
```

3. To confirm that the files were extracted, type one of the following commands and press ENTER:

➤ From a UNIX based command prompt, type:

```
ls -l
```

A listing similar to the following appears:

```
-rw-rw-rw- 1 user  pkware    1311 June 30 2:50 blue.fil
-rw-rw-rw- 1 user  pkware     131 June 30 2:50 green.doc
-rw-r--r-- 1 user  pkware   24360 June 30 3:30 test.zip
```

➤ From a DOS based command prompt, type:

```
dir
```

A listing similar to the following appears:

```
.          <DIR>      06-30-97  4:00a .
..         <DIR>      06-30-97  4:00a ..
TEST      ZIP        15,422  06-30-97  3:30a test.zip
GREEN     DOC         4,185  06-30-97  2:50a green.doc
BLUE      FIL         591    06-30-97  2:50a blue.fil
3 file(s)                20,198 bytes
2 dir(s)                 88,788,992 bytes free
```

## Extracting All Files From a .ZIP File

You can extract all of the files from your .ZIP file. When you extract specific files, as before, you must specify those files or file pattern. When you extract "all" files, you only have to type the name of the .ZIP file with the **extract** command. You can also, however, use the convention for specifying "all" files (\*).

## Tutorial K

Extract all the files from the test.zip file into the ext directory.

Follow these steps:

1. Verify that you are in the ext directory.
2. To extract all files from the test.zip file, type the following and press ENTER:

```
pkzip25 -extract test.zip
```

Because you are extracting files that already exist in your ext directory, PKZIP asks you if you want to overwrite the file by displaying the following prompt:

```
Overwrite (<Y>es/<N>o/<A>ll/ne<V>er/<R>ename/<Esc>)?
```

3. Type **"y"** for yes if you wish to overwrite the file.

The same prompt will appear for the other files that already exist in the EXT directory. If you wish to overwrite those files as well type **"y"**.

PKZIP extracts the files, and the screen will look similar to the following:

```
PKZIP(R) Version 2.50 FAST! Compression Utility
Copyright 1989-1997 PKWARE Inc. All Rights Reserved. Registered Version
PKZIP Reg. U.S. Pat. and Tm. Off. Patent No. 5,051,745

Extracting files from .ZIP: test.zip
Inflating: black.tut
PKZIP: (W7) warning! file: blue.fil already exists.
Overwrite (<Y>es/<N>o/<A>ll/ne<V>er/<R>ename/<Esc>)? y
Inflating: blue.fil
Inflating: brown.doc
Inflating: gold.tut
PKZIP: (W7) warning! file: green.doc already exists.
Overwrite (<Y>es/<N>o/<A>ll/ne<V>er/<R>ename/<Esc>)? y
Inflating: green.doc
Inflating: orange.fil
Inflating: pink.tut
Inflating: purple.txt
Inflating: red.txt
Inflating: tan.txt
Inflating: white.doc
Inflating: yellow.doc
```

4. To confirm that the files were extracted into your ext directory, type one of the following commands and press ENTER:

➤ From a UNIX based command prompt, type:

```
ls -l
```

A listing similar to the following appears:

```
-rw-rw-rw- 1 user  pkware  10900 June 30 2:50 black.tut
-rw-rw-rw- 1 user  pkware  1311  June 30 2:50 blue.fil
-rw-rw-rw- 1 user  pkware  1588  June 30 2:50 brown.doc
-rw-rw-rw- 1 user  pkware   619  June 30 2:50 gold.tut
-rw-rw-rw- 1 user  pkware   131  June 30 2:50 green.doc
-rw-rw-rw- 1 user  pkware   337  June 30 2:50 orange.fil
-rw-rw-rw- 1 user  pkware  10900 June 30 2:50 pink.tut
-rw-rw-rw- 1 user  pkware  1998  June 30 2:50 purple.txt
-rw-rw-rw- 1 user  pkware   315  June 30 2:50 red.txt
-rw-rw-rw- 1 user  pkware  3075  June 30 2:50 tan.txt
-rw-r--r-- 1 user  pkware  24360 June 30 3:30 test.zip
-rw-rw-rw- 1 user  pkware  17404 June 30 2:50 white.doc
-rw-rw-rw- 1 user  pkware   1637 June 30 2:50 yellow.doc
```

➤ From a DOS based command prompt, type:

```
dir
```

A listing similar to the following appears:

```
.          <DIR>      06-30-97  4:30a .
..         <DIR>      06-30-97  4:30a ..
TEST      ZIP        15,422  06-30-97  3:30a test.zip
GREEN     DOC          591    06-30-97  2:50a green.doc
BLUE      FIL         4,185  06-30-97  2:50a blue.fil
ORANGE    FIL          591    06-30-97  2:50a orange.fil
YELLOW    DOC         5,920  06-30-97  2:50a yellow.doc
RED       TXT         4,185  06-30-97  2:50a red.txt
PURPLE    TXT          591    06-30-97  2:50a purple.txt
WHITE     DOC         5,920  06-30-97  2:50a white.doc
BROWN     DOC         5,920  06-30-97  2:50a brown.doc
PINK      TUT         1,030  06-30-97  2:50a pink.tut
TAN       TXT         4,185  06-30-97  2:50a tan.txt
BLACK     TUT         1,030  06-30-97  2:50a black.tut
GOLD      TUT         1,030  06-30-97  2:50a gold.tut
13 file(s)                    50,600 bytes
 2 dir(s)                      88,727,552 bytes free
```

You have now extracted all of the files contained in the test.zip file.

## Overwriting Files that Already Exist in Your Directory

In the previous tutorial, you saw what appears when you extract files into a directory that contains file(s) with the same name. PKZIP offers several choices for handling the overwriting of files while extracting. Refer to *Chapter 4 - Extracting Files* on page 83 for complete information on extracting files from a .ZIP file.

## Extracting Files to a Specified Directory

The extraction tutorials up to this point had you extract the contents of the test.zip file into the current directory. With PKZIP, you can specify a "different" extract location for the .ZIP file - right in your command. Simply include the location, or directory path.

Tutorial L

Extract the file called red.txt, archived in the test.zip file, into the directory above (parent to) the tutorial directory.

Follow these steps:

1. Verify that you are in the tut directory. If you are still in the ext directory, change to the tut directory.
2. To extract the red.txt file archived in the test.zip file into the tutorial directory's parent directory, type the following command line and press ENTER:

```
pkzip25 -extract test.zip red.txt ..
```

The two periods (..) at the end of the command line tell PKZIP to extract the file into the directory up one level in the directory structure. PKZIP runs, and the following appears:

```
PKZIP(R) Version 2.50 FAST! Compression Utility
Copyright 1989-1997 PKWARE Inc. All Rights Reserved. Registered Version
PKZIP Reg. U.S. Pat. and Tm. Off. Patent No. 5,051,745
```

```
Extracting files from .ZIP: test.zip
Inflating: ../red.txt
```

As you can see from the "Inflating: ../" line above, PKZIP extracted the red.txt file to the parent directory of the current directory.

## Further Information on Extracting Files

The tutorials in the previous "extract" sections offer only a small sample of file extraction options. For example, you can extract "only" those files that are "newer" than the files in the extract directory. You can also determine how to handle the overwriting of files in the extract directory.

For information on these and other extract features, refer to *Chapter 4 - Extracting Files* on page 83.

## Understanding PKZIP Commands and Options

---

The previous sections demonstrated how to complete basic compression and extraction operations. In the tutorials, the **add** command was used to compress files while the **extract** command was used to extract files. These are the two most basic tasks in PKZIP.

With PKZIP, you can further customize compression and extraction tasks. You accomplish this by using additional commands and options. This section contains an introduction to these additional commands and options used in PKZIP, including basic conventions.

## Difference between a Command and Option

A "command" represents the main task you are performing. For example, the **add** command represents the task of "compressing files".

An "option" is a qualifier to the main task that customizes that task in some fashion. For example, to maximize compression, use the **maximum** option with the **add** command. To sort files that you are extracting, use the **sort** option.

Remember in Tutorial G when you used the **move** option with the **add** command? The **move** option acted as a "qualifier" to **add**, in that in addition to compressing the files, PKZIP also "moved" the files.

## Including an Option in Your Command Line

In your PKZIP command, an option usually appears immediately following the main command (for example, **add** or **extract**), separated by a space. The following is an example of using the **maximum** option with the **add** command, which instructs PKZIP to compress files at the "maximum level" but lowest speed:

```
pkzip25 -add -maximum test.zip white.doc
```

An "option" is usually preceded by a command that represents the main task you are performing.

Tutorial M

Compress the white.doc file into the test.zip file using the highest level of compression (the **maximum** option).

Follow these steps:

1. Verify that you are in the tut directory. If you are in the ext directory, change to the tut directory.

2. To compress files using the *maximum* option, type the following and press ENTER:

```
pkzip25 -add -maximum test.zip white.doc
```

PKZIP begins to compress the file using the 'maximum' or level 9 compression. Your screen will look similar to the following:

```
PKZIP(R) Version 2.50 FAST! Compression Utility
Copyright 1989-1997 PKWARE Inc. All Rights Reserved. Registered Version
PKZIP Reg. U.S. Pat. and Tm. Off. Patent No. 5,051,745

♦ Using compression level 9 - Maximum comp.

Updating .ZIP: test.zip
Updating File: white.doc   Deflating   (59.6%), done.
```

## Abbreviating Commands and Options

When you include an option in your PKZIP command, you can abbreviate that option, as long as it remains unique. In the preceding example, you could have typed "max" instead of "maximum" as it is the only option that begins with the letters "max". The same example would look like the following:

```
pkzip25 -add -max test.zip white.doc
```

## Combining Options

With PKZIP, you can combine options in the same command. For example, let's say you want to compress files at the maximum level of compression, in addition to adding a comment to the header area of the .ZIP file. To accomplish this, you would use the *header* and *maximum* options in your command, as in the following example:

```
pkzip25 -add -header -max test.zip brown.doc
```

The order in which you type the options is not important. You could have typed:

```
pkzip25 -add -max -header test.zip brown.doc
```

Not all options can be combined with others. For example, in the previous example, you wouldn't use maximum with another compression level option because you can only select one compression level. See *Appendix A* on page 119 for a complete list of commands and options and their functionality

## Tutorial N

In this tutorial, you will compress the brown.doc file into the test.zip file using the **fast** option. Additionally, you will add a header comment to the test.zip file using the **header** option. The **fast** option instructs PKZIP to use the fastest compression speed.

Follow these steps:

1. Verify that you are in the tut directory.
2. To compress files using the **fast** and **header** options, type the following and press ENTER:

```
pkzip25 -add -fast -header test.zip brown.doc
```

The **fast** option emphasizes faster compression speed over the compressed file size. The **header** option instructs PKZIP to prompt you to enter a header comment that will then appear in the header area of the .ZIP file.

PKZIP will display the following prompt:

```
Zip Header ?
```

3. Specify a header comment and press ENTER. The PKZIP output screen, including the " Zip Header ?" prompt, will look similar to the following:

```
PKZIP(R) Version 2.50 FAST! Compression Utility
Copyright 1989-1997 PKWARE Inc. All Rights Reserved. Registered Version
PKZIP Reg. U.S. Pat. and Tm. Off. Patent No. 5,051,745

♦ Using compression level 2 - Fast
Updating .ZIP: test.zip
Zip Header ? this is my header
Adding File: brown.doc Deflating (58.8%), done.
```

As you can see from the screen output, PKZIP is using compression level 2. Also note that we supplied the comment "this is my header" at the "Zip Header ?" prompt.

## Commands and Options That Contain Values

Some commands and options contain specific values or sub-options that further qualify a compression or extraction operation. For example, PKZIP contains an option called **level** that allows you to specify the level of compression and speed you want to use. The sub-option or value would equal any digit from 0 through 9, with 0 being no compression at the fastest speed, and 9 being the most compression at the slowest speed

When you include a value with a command or option, the value is separated from the option with an equal sign. For example, if you wished to specify minimum compression while emphasizing maximum speed, you could use the following option and sub-option:

**level=1**

An example of a command that uses values is the **add** command. For example, you can compress only files that exist in the .ZIP file and that have changed since the last time you compressed them. This value is called "freshen" and you include it with your add command as follows:

**add=freshen**

Some commands and options contain more than one value from which to choose. They also contain "default" values, which are listed via a PKZIP option called **configuration**. For information on values and defaults, refer to *Chapter 6 - Changing Defaults Using the Configuration File* on page 107.

For complete information on the values available with commands and options, refer to *Appendix A: PKZIP Options: A Quick Reference* on page 119.

## Tutorial 0

In this tutorial, you will compress files using a command and option, both of which contain sub-options or values. Specifically, you will compress only files that exist in the test.zip file, but that have changed. Additionally, you will specify that the file be minimally compressed while emphasizing maximum compression speed. This is accomplished by using the **add=freshen** command as well as the **level=1** option.

Follow these steps:

1. Verify that you are in the tut directory.
2. Open a text editor (e.g. vi; notepad; e). Type anything in the edit screen that you wish. Save the file into your tut directory as test.txt. Exit the text editor. A file called test.txt should now exist in your tut directory.
3. Add the test.txt file to the test.zip archive by typing the following:

```
pkzip25 -add test.zip test.txt
```

The test.txt file is added to the test.zip archive.

4. Re-open test.txt file located in the tut directory with your text editor (e.g. vi; notepad; e). Edit or add characters in the edit screen. Re-save the file as test.txt into your tut directory. An updated test.txt file should now exist in your tut directory.
5. Type the following and press ENTER:

```
pkzip25 -add=freshen -level=1 test.zip
```

PKZIP adds only those files that have been updated since test.zip was first created or last modified. Since we modified the test.txt file in Step 4, it will be updated in the test.zip archive. In addition, our command line tells PKZIP to compress the files using minimal compression while emphasizing maximum compression speed.

The screen output will look similar to the following:

```
PKZIP(R) Version 2.50 FAST! Compression Utility
Copyright 1989-1997 PKWARE Inc. All Rights Reserved. Registered Version
PKZIP Reg. U.S. Pat. and Tm. Off. Patent No. 5,051,745

♦ Using compression level 1 - Maximum speed

Freshening .ZIP: test.zip
Updating File: test.txt Storing ( 0.0%), done.
```

## Commands and Options: Compression or Extraction?

Some command and options in PKZIP apply only to "compressing" while others apply only to "extracting". Some of these commands and options apply to both.

Refer to the next two chapters, *Compressing Files* and *Extracting Files*, for information on which options you can use with which tasks.

## Setting Defaults

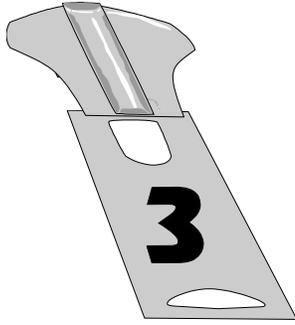
With PKZIP, you can set up and change default values for many of the commands and options. *Chapter 6 - Changing Defaults Using the Configuration File* on page 107 shows you how to set, view, and edit PKZIP's default settings.

## What's Next

---

Congratulations! You have learned enough information to use the basic features of PKZIP. Nevertheless, this chapter is only the beginning. To learn more features, please refer to the remaining chapters in this manual.





## Compressing Files

### Overview:

This chapter contains information on the commands and options used in file compression and archival.

### Sections in This Chapter:

- Introduction
- Compressing New and Existing Files
- Compressing Files in Subdirectories
- Storing Directory Path Information
- Compressing Files With a LIST FILE
- Compressing Files Based on Some Criteria
- Compressing Files in a Specific Format
- Storing File Attribute Information
- Including Additional Information in .ZIP File
- Setting the Compression Level
- Sorting Files Within a .ZIP File
- Moving Files to a .ZIP File
- Creating Self-Extracting .ZIP Files
- Compressing Files to Diskette

## Introduction

---

PKZIP offers several options for compressing files into a .ZIP file. The previous chapter introduced you to some PKZIP basics. This chapter contains detailed information on the features and compression functionality presently available in PKZIP.

## Conventions in This Chapter

Most commands and options discussed in this and subsequent chapters work for all versions of PKZIP. In instances where a command or option is specific to a platform or operating system, the manual will indicate as much. (e.g. UNIX, DOS, OS/2).

**Note:** The **(DOS)** specification refers to *any* DOS based shell including MS-DOS shells running on Windows 95 and Windows NT.

In some of the headings in this chapter, a word appears immediately below the heading. That word represents the actual command or option you would type in your PKZIP command line (and in many cases, a value or suboption in addition to the command e.g. **add=all**). If the text under the heading is the command followed by an equal sign and another word, it means that the other word is a value (sub-option) that goes with that command.

## Default Values for Commands and Options

For each compression task in this chapter, the command or option that represents that task contains a default value. A default value represents the action that occurs when only the name of the command or option is included in your PKZIP command. For example, the default for the **add** command is to compress "all" files.

For commands and options that contain defaults, these defaults are stored in the Configuration file. PKZIP's Configuration Options can *only* be edited through PKZIP. Refer to *Chapter 6 - Changing Defaults Using the Configuration File* on page 107 for more information.

## Compressing New and Existing Files

---

When you compress files into a .ZIP file, you can create a new .ZIP file or add/update files in an existing .ZIP archive. For example, you might only want to compress files that end in ".doc". On the other hand, you might want to update files in an existing .ZIP file, but only those that have changed since the last time you compressed them.

**Note:** The *add* command is used for creating new .ZIP files as well as adding files to an existing .ZIP archive.

## Compressing All Files in a Directory

### *add*

You have the option of compressing all files in a particular directory with a single command. To do this, you do not have to specify each file. Simply type *pkzip25 -add*, and the name of your .ZIP file, as shown below:

```
pkzip25 -add test.zip
```

In this example, all files in the current directory will be compressed into the test.zip file.

**Note:** The above command only compresses files that are located in that particular directory, not in the subdirectories that might appear below that directory. To learn how to compress files that appear in subdirectories, refer to the *Compressing Files in Subdirectories* section on page 48.

You can also specify files from a different directory if you wish. For example, if you were in a parent directory to a directory called temp and you wanted to compress all the files in the temp directory, you could type the following:

```
pkzip25 -add test.zip temp/*
```

The resultant test.zip file is stored in the current directory (the parent directory to the temp directory in our example).

**Note:** By default, all files in a specified directory are added to your .ZIP file with PKZIP's **add** command. Unless you've modified the **add** default setting through the PKZIP Configuration Settings file, you wouldn't need to include the 'all' suboption in your command line to compress all files. For more information on modifying PKZIP's default values through the PKZIP Configuration Settings file, see *Chapter 6 - Changing Defaults Using the Configuration File* on page 107.

## Compressing New and Modified Files

### ***add=update***

PKZIP allows you to specify that only new or modified files are added to an existing .ZIP archive. When the **update** sub-option is used in conjunction with a `pkzip25` command, the files specified for archiving will be compared against the files already present in the .ZIP file. If the file(s) to be added into the .ZIP file is already present and is *not* newer, PKZIP will *not* re-compress the file.

By using this option, you may save yourself time when archiving files that are backed up repeatedly. This option differs from the behavior of the **freshen** option in that files which are not already present in the .ZIP file will be added.

To compress only updated files or files not already archived in a specific .ZIP file, use the **update** sub-option with the **add** option, as shown below:

```
pkzip25 -add=update test.zip *.doc
```

In this example, a .ZIP file called `test.zip` is created in the current directory. All files in the current directory matching the file specification (`*.doc`) will be added or updated into the `test.zip` archive.

## Compressing Only Files That Have Changed

### ***add=freshen***

The ***freshen*** value allows you to selectively update files archived in a .ZIP file. PKZIP will compress only files that exist in the .ZIP file and that have changed. To update files that have changed, use the ***freshen*** value with the ***add*** option, as shown below:

```
pkzip25 -add=freshen test.zip
```

You can also abbreviate the value, so you could type the following instead:

```
pkzip25 -add=fre test.zip
```

When you use ***freshen*** with ***add***, only files that already exist in the .ZIP file "and" that have also changed will be compressed. No new files will be added to the .ZIP file.

If you only want to re-compress specific files, simply include those files in your command. For example, if you wanted to re-compress a file called resume.doc, you would type something like this:

```
pkzip25 -add=freshen test.zip resume.doc
```

In the above example, only resume.doc will be re-compressed into the test.zip file. This assumes that the version of resume.doc being added is newer than the version of resume.doc that already exists in the .ZIP file.

## Clearing Archive Attributes (DOS, OS/2)

### ***add=incremental***

If you wish to add files to a .ZIP file that have the archive attribute set and subsequently clear the archive attribute on those files, use the ***add*** command with the ***incremental*** sub-option. If you wish to add files to a .ZIP file that have the archive attribute set and *not* clear the archive attribute on those files, use the ***add*** command with the ***-incremental*** sub-option.

The *incremental* and *-incremental* sub-options can be very useful when backing up files. If, for example, the *incremental* sub-option is specified, only files with the archive attribute will be compressed, and the archive attribute will be set to OFF when the ZIP operation is complete for these files.

In the following command line example, PKZIP will add only those files to test.zip with the archive attribute set. Additionally PKZIP will clear the archive attribute on any of the source files that have been added to test.zip.

```
pkzip25 -add=incremental test.zip
```

The next time you run this command, only those files that have the archive attribute set (new or updated files) will be added to the test.zip file.

### **Archive Attribute Explained**

Any given file may have several properties associated with it. One such property or attribute is called the Archive Attribute. When a file is created, this attribute is set to be ON. In addition, if a file is altered, the attribute is set. After a file has been backed-up by a program which uses this attribute, the attribute is switched off. By making use of the archive attribute, you can make certain that you get all files that are new or changed. You save time by not backing up files you have previously archived. This process is called an Incremental Backup.

## **Compressing Files in Subdirectories**

---

### ***recurse***

PKZIP does not automatically compress files that appear in subdirectories, unless you specify those directories, or use the *recurse* option with the *add* command. With the *recurse* option, all specified files in a directory structure, including files located in subdirectories will be compressed.

For example, let's assume that you have a directory called `tut` with a nested subdirectory called `test`. To compress all of the files in the `tut` directory and all files in the `tut/test` directory, you would type the following in the `tut` directory:

```
pkzip25 -add -recurse test.zip *
```

All files in the `tut` directory as well as those files in subdirectories of the `tut` directory are compressed. However, directory path information is not stored within the `.ZIP` file. If you want to store directory information within your `.ZIP` file (in addition to compressing all the files in those directories), use the *path* option with the *recurse* option or simply use the *directories* option.

**Note:** UNIX users should utilize the *include* option or place quotation marks around wildcard designations to bypass automatic wildcard expansion by your shell, which may restrict your pattern search. See the **Getting Started Manual** for more information.

## Storing Directory Path Information

---

### *path*

Normally, when PKZIP compresses files, only the files are stored within the `.ZIP` file, not the paths of those files. However, you can instruct PKZIP to store the directory path information of a file within the `.ZIP` file. (When you extract the files later, you have the option of retaining the directory structure of those files.)

If for example, a file you are compressing appears in the `doc/temp` directory, you can store the file within the `.ZIP` file as:

```
doc/temp/<filename>
```

For example, let's assume that you are currently in a directory called PKWARE. Underneath the PKWARE directory is a nested subdirectory called test. If you wish to store relative directory path information within a .ZIP file, use the **path** option with the **add** command:

```
pkzip25 -add -path test.zip test/*.txt
```

In the previous example, any .txt files that exist in the /PKWARE/test directory are archived in the test.zip file. Additionally, the relative path (test/\*.txt) is also stored in the test.zip. If the files stored in test.zip were then extracted, using the **directories** option, this path information would be preserved and the files, as well as the directory where the files were originally stored, would be created and/or extracted. Keep in mind that with the **path** option alone, PKZIP will only search the current directory for files. If you wish to search subdirectories as well as preserve the directory structure, use the **path** option in conjunction with the **recurse** option or simply use the **directories** option.

## **Additional Methods for Storing Directory Path Information**

In addition to storing relative path information, PKZIP allows you to further customize path information storage in your .ZIP files. Several sub-options allow you specify exactly what directory and path information is to be stored.

Each sub-option is a value that you include with the path option. The path option and/or sub-option will override the path value in the PKZIP Configuration File. If no sub-option is specified, only relative path information is stored. Examples of each sub-option appear in the table below:

Sub-option:	To:	For example:
<b>current</b>	Store the directory path relative to the current location.	<p>pkzip25 -add -path=current docs.zip docs/*</p> <p>In this example, only directory information under the docs directory will be stored. Parent directory information will not be stored.</p>
<b>root or full</b>	Store the full path, starting from the root directory down.	<p>pkzip25 -add -path=root docs.zip docs/*</p> <p>In this example, the entire directory path, starting from "root" directory will be stored.</p>
<b>specify</b>	Store the directory path information that is specified in your PKZIP command.	<p>pkzip25 -add -path=specify docs.zip temp/docs/*</p> <p>In this example, temp/docs is the directory information that will be stored.</p>
<b>relative</b>	Store the directory path relative to the current working directory of the drive specified. <b>(DOS)</b>	<p>pkzip25 -add -directories=relative docs.zip c:*.doc z:*.doc</p> <p>In this example the path information for those directories recursed under the current working directory (for both the C: and Z: drives) will be stored.</p>
<b>none</b>	Turn off the path option. (Used to override configuration file).	<p>pkzip25 -add -path=none docs.zip /temp/docs/*</p> <p>In this example, only the file names are stored.</p>

## Storing and Recreating Directory Path Information

### ***directories***

The ***directories*** option works with both the ***add*** and ***extract*** commands. When used with the ***add*** command, the ***directories*** option is equivalent to using both the ***recurse*** and ***path*** options together. The ***directories*** option instructs PKZIP to search subdirectories for files and save the files with their directory path information in the .ZIP file. When used with the ***extract*** command, the ***directories*** option extracts the files while preserving the directory tree structure and file location.

The following example illustrates compression operations using the ***add*** command with the ***directories*** option. The command line that follows will compress a file called resume.doc into a file called test.zip. resume.doc, for the purposes of this example, resides in the temp directory.

```
pkzip25 -add -directories test.zip resume.doc
```

The screen output will look similar to the following:

```
PKZIP(R) Version 2.50 FAST! Compression Utility
Copyright 1989-1997 PKWARE Inc. All Rights Reserved. Registered Version
PKZIP Reg. U.S. Pat. and Tm. Off. Patent No. 5,051,745

Creating .ZIP: test.zip
Adding File: temp/resume.doc Deflating (69.3%), done.
```

As you can see from the Adding File: line above, the current path to the file resume.doc is listed (temp/resume.doc). For information on extracting files compressed with directory information, see the *Retaining Directory Structure while Extracting* section on page 92.

**Note:** UNIX users should utilize the ***include*** option or place quotation marks around wildcard designations to bypass automatic wildcard expansion by your shell, which may restrict your pattern search. See the **Getting Started Manual** for more information.

As with the path option, PKZIP provides several methods for storing directory path information. The available sub-options are listed in the following table:

Sub-option:	To:	For example:
<b>current</b>	Store the directory path relative to the current location.	<p>pkzip25 -add -directories=current docs.zip docs/*</p> <p>In this example, only directory information under the docs directory will be stored. Parent directory information will not be stored.</p>
<b>root or full</b>	Store the full path, starting from the root directory down.	<p>pkzip25 -add -directories=root docs.zip docs/*</p> <p>In this example, the entire directory path, starting from "root" directory will be stored.</p>
<b>specify</b>	Store the directory path information that is specified in your PKZIP command.	<p>pkzip25 -add -directories=specify docs.zip temp/docs/*</p> <p>In this example, temp/docs is the directory information that will be stored.</p>
<b>relative</b>	Store the directory path relative to the current working directory of the drive specified. <b>(DOS)</b>	<p>pkzip25 -add -directories=relative docs.zip c:*.doc z:*.doc</p> <p>In this example, the path information for those directories recursed under the current working directory (for both the C: and Z: drives) will be stored.</p>
<b>none</b>	Turn off the path option. (Used to override configuration file).	<p>pkzip25 -add -directories=none docs.zip /temp/docs/*</p> <p>In this example, only the file names are stored.</p>

Refer to the previous section, *Additional Methods for Storing Directory Path Information*, for more information.

## Compressing Files with a LIST File

---

Rather than specifying a specific file or file pattern in your command line, you have the option of pointing to a LIST file in your command line. The LIST file is an ASCII text file that contains file names and their locations. A LIST file can be an ideal solution for users who archive specific file sets on a regular basis. It saves time in that it is not necessary to type appropriate file names and paths each time you wish to compress these files with PKZIP. A LIST file may contain wild card specifications (\*,?) as well as exact file names and paths.

A LIST file in a DOS based environment might look similar to the following:

```
*.exe
*.doc
\tut\*.doc
\tut\?????.*
pkzip.html
```

A LIST file in a UNIX based environment might look similar to the following:

```
/usr/local/bin/pkware/pkzip25/*.doc
/usr/local/bin/pkware/pkzip25/pkzip.html
/usr/local/bin/pkware/pkzip25/?????.exe
/*
```

Using the @ character in your command line, you can point to the LIST file. We'll assume that the LIST file is called lst.txt and that it resides in your current directory. An example of such a command line follows:

```
pkzip25 -add test.zip @lst.txt
```

In the previous example, PKZIP creates a file called test.zip using file information it retrieves from a file called lst.txt, located in the current directory. lst.txt contains file location information that PKZIP uses to add files to the test.zip archive. You may also use a LIST File to specify files for exclusion from an archive, based on some criteria, using the **exclude** option. The **exclude** option is discussed in more detail on page 58. Use the **listchar** option to specify something other than the @ character as the LIST character. For more information on the **listchar** option, see *Chapter 7 - Command Characteristics* on page 115.

**Note:** The file format for a LIST file when extracting files may differ from the format referenced above. See the *Extracting Files with a LIST File* section on page 94 for more information.

## Compressing Files Based on Some Criteria

---

### ***after, before, include, exclude***

With PKZIP, you can compress files based on certain criteria. You can compress:

- Files that are newer than a specified date.
- Files that are older than a specified date.
- All files of a specific pattern.
- All files "except" those specified.

Refer to the sections that follow for more information.

### Compressing Files Newer Than a Specified Date

#### ***after***

With PKZIP, you can choose to compress only those files that are newer or equal to a specified date. To do this, use the ***after*** option, followed by an equal sign and the date, as shown below:

```
pkzip25 -add -after=062495 test.zip
```

In this example, only files that contain a date newer or equal to June 24, 1995 will be compressed.

With PKZIP, you can enter dates in the following formats:

- mmddyy
- mmddyyyy

The order in which you enter the month, date, and year depends on your *locale* setting. For more information on the *locale* setting, see *Chapter 7 - Command Characteristics* on page 114.

## Compressing Files Older Than a Specified Date

### *before*

With PKZIP, you can choose to compress only files that are older than a specified date. To do this, use the *before* option, followed by an equal sign and the date, as shown below:

```
pkzip25 -add -before=062495 test.zip
```

In this example, only files that contain a date older than June 24, 1995 will be compressed.

With PKZIP, you can enter dates in the following formats:

- mmddyy
- mmddyyyy

The order in which you enter the month, date, and year depends on your *locale* setting. For more information on the *locale* setting, see *Chapter 7 - Command Characteristics* on page 114.

## Including Files That Match a Pattern

### *include*

PKZIP allows you to compress files that match a specific pattern, for example, all files that end in the .doc extension.

To compress files that match a pattern, simply include the pattern after the *add* command, as in the following example:

```
pkzip25 -add test.zip *.doc
```

You have the option of specifying a default file pattern setting in your PKZIP Configuration Settings file. If, for example, you want to automatically include all files with the extension of .doc in PKZIP compress and extract operations, enter the following:

```
pkzip25 -config -include=*.doc
```

When you use the **include** option with the **configuration** command, PKZIP prompts you to configure the **include** default for add *and/or* extract operations. The .doc file pattern will henceforth be included by default in compress *and/or* extract operations. If, for example, you type the following command line:

```
pkzip25 -add test.zip *.txt
```

Those files with .txt *and* .doc extensions are added to the test.zip archive.

If you set up a default pattern in the Configuration file, but want to specify a different pattern for this instance only (for example, compress all files that end in .txt), use the **include** option followed by an equal sign, then the file pattern, as in the example below:

```
pkzip25 -add -include=*.txt test.zip
```

If you do not set a default for **include** in the Configuration file, it's not necessary to specify the **include** option in your command, only the file pattern.

For more information on modifying PKZIP's default values through the PKZIP Configuration Settings file, see *Chapter 6 - Changing Defaults Using the Configuration File* on page 107.

## Excluding Files from Being Compressed

### ***exclude***

When you compress files, you might want to exclude specific files from being compressed or extracted, for example, all files that end in .bmp. To exclude files, use the ***exclude*** option with the ***add*** command, as shown below:

```
pkzip25 -add -exclude=*.bmp test.zip
```

In the example above, all files except those that end in the .bmp extension will be compressed.

Furthermore, you might want to exclude a list of files from being compressed or extracted, for example, all files that exist in the list file lst.txt. To exclude files listed in a list file, use the ***exclude*** option with the ***add*** command, as shown below:

```
pkzip25 -add -exclude=@lst.txt test.zip
```

In the example above, all files except those listed in the lst.txt file will be compressed.

You have the option of specifying a default file pattern setting in your PKZIP Configuration Settings file. If, for example, you want to automatically exclude all files with the extension of .doc in PKZIP compress and extract operations, enter the following:

```
pkzip25 -config -exclude=*.doc
```

When you use the ***exclude*** option with the ***configuration*** command, PKZIP prompts you to configure the ***exclude*** default for add *and/or* extract operations. The .doc file pattern will henceforth be excluded by default in compress *and/or* extract operations.

## Compressing Files in a Specific Format

---

### ***shortname***

The ***shortname*** option allows you to convert a .ZIP file name in *long* file name format to a DOS equivalent *short* (8+3) file name before compressing the file(s). You may specify how you wish files to be compressed by using the ***shortname*** option with one of the following sub-options:

Sub-option:	To:	For example:
<b>dos</b>	Convert .ZIP file to a DOS-equivalent short file name.	pkzip25 -add -short=dos save.zip
<b>os2</b>	Convert .ZIP file to a OS/2-equivalent short file name.	pkzip25 -add -short=os2 save.zip

**Note:** Windows NT uses the DOS file name from the system but generates an OS/2 file name internally. PKZIP includes sub-options for both OS/2 and DOS because of the differences in the way each operating system handles short file names.

## Storing File Information

---

PKZIP allows you to store specific file attribute/information within your .ZIP file. You can:

- Store file attributes, including hidden, system, and read-only.
- Store extended file attribute information.
- Remove (mask) file attributes.

Refer to the sections that follow for more information.

## Compressing Files That Contain Certain Attributes (DOS, OS/2)

### ***attributes***

PKZIP allows you to compress files based on the attributes that they possess. These attributes are usually assigned either by the creator of a file, a system administrator, or by the operating system. The following are attributes you can store:

- hidden
- system
- read only

The default attribute for purposes of compression is "read-only". That is, if you do not use the ***attributes*** option on your command line, "system" and "hidden" files are not compressed into your .ZIP files. If you do use the ***attributes*** option on your command line but do *not* specify a sub-option, all files, including those that include "hidden" and "system" attributes, are compressed into your .ZIP files.

To specify a file attribute, you must include it with the ***attributes*** option in your command line. Each attribute is a "value" for the ***attributes*** option. You can:

- Specify which file attributes to compress.
- Override values in the Configuration file.
- Turn off the ***attributes*** option.

The table below lists all of the available sub-options for storing file attribute information:

Sub-Option:	To:	For example:
<b>hidden</b>	Compress files including those that contain the "hidden" file attribute.	<code>pkzip25 -add -attributes=hid test.zip</code>
<b>system</b>	Compress files including those that contain the "system" file attribute.	<code>pkzip25 -add -attributes=sys test.zip</code>
<b>readonly</b>	Compress files including those that contain the "read-only" file attribute.	<code>pkzip25 -add -attributes=read test.zip</code>
<b>all</b>	Compress files including those that contain the hidden, system, or read-only file attribute.	<code>pkzip25 -add -attributes=all test.zip</code>
<b>none</b>	Turn off the attributes option in the Configuration file or compress files that do not have any attributes set.	<code>pkzip25 -config -attributes=none</code>

You may use a dash (-) before an **attributes** sub-option on your command line to exclude files with a specific attribute from being added regardless of the default attributes configuration setting. If, for example, the default attributes configuration setting was set to "all", you could enter the following command line to exclude hidden files from being added to the test.zip file.

```
pkzip25 -add -attributes=-hidden test.zip
```

## Compressing Files Based on File Type (UNIX)

### *filetype*

PKZIP allows you to process files based on a specific file type. Use the *filetype* option with the following sub-options to process specific file types:

Sub-Option:	To:	For example:
<b>hidden</b>	Include/exclude hidden files. This would include filenames that have a dot (.) in the first position of the filename. (e.g. .profile)	pkzip25 -add -filetype=hid test.zip
<b>pipe</b>	Include/exclude pipe files. This would include files with a mode that begins with a "p". (e.g. prwxr-xr-x)	pkzip25 -extract -filetype=pipe test.zip
<b>slink</b>	Include/exclude symbolically linked files. This would include files with a mode that begins with a "l". (e.g. lrwxr-xr-x)	pkzip25 -add -filetype=slink test.zip
<b>hlink</b>	Include/exclude "hard" linked files. Hard linked files have a link count greater than one.	pkzip25 -extract -filetype=hlink test.zip
<b>none</b>	Exclude all file types except for those specified on the command line. The none option is typically followed by one or more file types. Only the specified types are included in the .ZIP file. For example, filetype=none,pipe results in only PIPE files being included.	pkzip25 -config -filetype=none,slink
<b>all</b>	Include/exclude all file types.	pkzip25 -add -filetype=all test.zip
<b>regular</b>	Include/exclude regular files. This is the default setting. If no filetype is specified, PKZIP will automatically include/exclude regular files.	pkzip25 -extract -filetype=regular test.zip

**Note:** A "-" before a *filetype* sub-option tells PKZIP to exclude the specified filetype(s) regardless of the default configuration setting. (e.g. -filetype=-hidden will exclude hidden files regardless of the default configuration setting)

## Extended Attribute Storage

### *noextended*

When PKZIP adds files to an archive, it automatically stores extended attributes with those files. PKZIP defines extended attributes as those file attributes not defined by the standard FAT file system attributes (Read-Only, Archive, System, Hidden, Volume, Directory). Extended attributes normally represent a characteristic of a file. For example, PKZIP may store the date and time a file was last modified as an extended attribute.

If, however, you do *not* wish to store or extract extended attribute information, use the *noextended* option. The following command line example uses the *noextended* option with *add* command.

```
pkzip25 -add -noextended test.zip readme.doc
```

PKZIP will run and display the following:

```
PKZIP(R) Version 2.50 FAST! Compression Utility
Copyright 1989-1997 PKWARE Inc. All Rights Reserved. Registered Version
PKZIP Reg. U.S. Pat. and Tm. Off. Patent No. 5,051,745

◆ Excluding Extended Attributes

Creating .ZIP: test.zip
Adding File: readme.doc Deflating (26.8%), done.
```

**Note:** The *noextended* option does not effect the storage of the offline, temporary, and system attributes on DOS systems. Similarly, the *noextended* option does not effect the storage of filetype attributes on UNIX systems.

### Extended Attributes and the OS

Extended attributes are automatically added to .ZIP archives when they are created. PKZIP does not display a message indicating that it is saving extended attributes. However, be aware that PKZIP running on a UNIX system stores different extended attributes than PKZIP running on a Win32 system. The following table lists the extended attributes that PKZIP stores relative to the UNIX and Win32 operating systems:

UNIX	Win32
user ID	create time
group ID	last modification time
last modification time	last access time
last access time	
link information	

**Note:** Typically, PKZIP will automatically extract extended attributes with archived files and/or directories. Consequently, PKZIP will overwrite existing files, directories and extended attributes (EAs) with those files, directories, and extended attributes (EAs) stored in the .ZIP file. It should be noted, however, that extended attribute preservation is dependent on such things as the user's file system privileges as well as the option=suboption (e.g. *id*, *permission*, *times*) specified on the command line or in the configuration file.

## Extended Attributes and 204g Compatibility

### **204**

By default, PKZIP does not enable PKZIP for DOS 2.04g compatibility. When 204g compatibility is enabled, extended attribute data is stored in both the Local header and Central header records. This will result in a slightly larger .ZIP file size, but improves the chance that extended attribute information can be recovered if the .ZIP file should become damaged. It also ensures the extended attribute information is always retained if the file is generated with a version of PKZIP other than 2.04g. This option is ignored when extracting. The **204** option also limits the number of files that can be added to a .ZIP archive from 65,535 to 16,383. To enable 204g compatibility, use the **204** option as in the following example:

```
pkzip25 -add -204 test.zip *
```

## Removing File Attributes when Compressing (DOS, OS/2)

### **mask**

By default, PKZIP will compress all normal and read-only files. If, however, you choose to compress files with other attributes (system, hidden) using the **attribute** option, you can use the **mask** option to remove those same attributes on compression or extraction. The **mask** option allows you to retain (or not retain) specific file attributes within that .ZIP file.

The table below lists all of the available sub-options for masking file attribute information:

Sub-Option:	To:	For example:
<b>hidden</b>	Remove the hidden file attribute from files.	<code>pkzip25 -add -mask=hidden test.zip *</code>
<b>system</b>	Remove the system file attribute from files.	<code>pkzip25 -add -mask=system test.zip *</code>
<b>readonly</b>	Remove the read-only file attribute from files.	<code>pkzip25 -add -mask=readonly test.zip *</code>
<b>none</b>	Turn off file masking.	<code>pkzip25 -add -mask=none test.zip *</code>
<b>all</b>	Remove all attributes from files.	<code>pkzip25 -add -mask=all test.zip *</code>

The mask sub-options can be used on your command line individually or as a comma-separated list.

You may use a dash (-) before a mask sub-option on your command line to preserve a file attribute being added or extracted with a file, regardless of the default mask configuration setting. If, for example, the default mask configuration is set to "all" you could enter the following command line to preserve the hidden attribute associated with those files (with the hidden attribute set) being added to the test.zip file.

```
pkzip25 -add -mask=-hidden test.zip
```

**Note:** By default, PKZIP will not mask any file attributes when compressing files. All file attributes will be retained in the .ZIP file. By default, PKZIP will mask all file attributes when extracting files. If file attributes have been stored in a .ZIP archive, they will not be restored upon file extraction.

## Including Additional Information in a .ZIP File

---

With PKZIP, you can include additional information in your .ZIP file, such as a "comment", to identify that .ZIP file.

You can include a:

- Text comment.
- Password to protect your .ZIP file.
- Header comment.
- Volume label.
- Date for the .ZIP file (other than the creation date).

Refer to the sections that follow for more information.

### Including a Text Comment

#### ***comment***

With PKZIP, you can include a comment for the individual files within a .ZIP file. There are several options for adding comments to your .ZIP files. To include a comment, use the ***comment*** option alone or with the ***add*** command. When you type the command, PKZIP prompts you to enter the comment.

The table below lists the available sub-options for adding comments to your .ZIP archives:

Sub-Option:	To:	For example:
<b>all</b>	Comment all of the files and any new files added.	pkzip25 -add -comment=all test.zip *
<b>unchanged</b>	Comment only files existing in the ZIP file that are not either updated or being added.	pkzip25 -add -comment=unchanged test.zip *
<b>add</b>	Comment only the new files added.	pkzip25 -add -comment=add test.zip *
<b>none</b>	Disable the comment option.	pkzip25 -add -comment=none test.zip *
<b>freshen</b>	Comment all of the files updated in the ZIP file.	pkzip25 -add -comment=freshen test.zip *
<b>update</b>	Comment all files added and updated in the zip file.	pkzip25 -add -comment=update test.zip *

**Note:** Comment length is limited to 59 characters.

## Including a Password

### ***password***

With PKZIP, you can include a password with your .ZIP file to restrict access to that file.

To include a password, use the ***password*** option with the ***add*** command, and do one of the following:

- Type the password (preceded by an equal sign) as part of your command. For example:

```
pkzip25 -add -password=<type password here> test.zip
```

- If you include the password option only, PKZIP will prompt you to enter a password. For example:

```
pkzip25 -add -password test.zip
```

When you press ENTER, the following prompt appears:

```
password?
```

Type your password, which will appear on your screen as asterisks for privacy. Press ENTER to save your password. PKZIP asks you to confirm the password by displaying the following:

```
Re-enter password for verification.  
password?
```

Re-type your password and press ENTER. If the password you enter matches the one you originally entered, PKZIP will start to compress your files, and the operating system prompt will appear. If the passwords do not match, PKZIP displays the following:

```
Passwords don't match! Please try again.  
password?
```

Re-type the correct password.

To extract a password protected file, use the **password** option with the **extract** command.

- Type the password (preceded by an equal sign) as part of your command. For example:

```
pkzip25 -extract -password=secret test.zip
```

If you've specified the correct password, the files will be extracted to the current directory. If the password you've entered is incorrect, you will see a warning message similar to the following:

```
PKZIP: (w20) warning! Incorrect password for file: filename.ext
```

Retype your command line with the correct password.

- If you include the password option only, PKZIP will prompt you to enter a password. For example:

```
pkzip25 -extract -password test.zip
```

When you press ENTER, the following prompt appears:

```
Password?
```

Type your password, which will appear on your screen as asterisks for privacy. Press ENTER. If you've specified the correct password, the files will be extracted to the current directory. If the password you've entered is incorrect, you will see a warning message similar to the following:

```
PKZIP: (w20) warning! Incorrect password for file: filename.ext
```

Retype your command line and when prompted enter the correct password.

**Note:** Passwords are case sensitive.

## Including a Header Comment

### ***header***

With PKZIP, you can include a general comment for a .ZIP file. This is called a "header" comment because it appears in the header portion of a .ZIP file. This differs from the ***comment*** option in that the "header" comment applies to the entire .ZIP file, not to individual files within the .ZIP file.

**Note:** Headers for .ZIP files are limited to 16K in size. PKZIP will automatically truncate headers larger than 16K.

To include a header comment, use the ***header*** option and the comment or comment file with the ***add*** command. PKZIP provides several methods to include the comment. You can:

- Include an existing file as the header. With this method, you type the ***header=@filename.ext*** option. If there are no spaces in the file name, it is not necessary to use quotation marks. For example:

```
pkzip25 -add -header=@header.txt test.zip *
```

- Type the actual comment as part of the command. With this method, you include an equal sign, followed by the comment. If there are no spaces in your comment, it is not necessary to use quotation marks. Our example comment does include spaces so therefore our command line would look like the following:

```
pkzip25 -add -header="This is the comment" test.zip *
```

## Including a Volume Label (DOS, OS/2)

### ***volume***

With PKZIP, you can include a volume label with your .ZIP file. This can be any drive letter, even if it is not the same drive from which the files were compressed. For example, if you compressed files from the C:\ drive, you could store the volume label from your D:\ drive as your volume label.

To include a volume label, use the **volume** option with the **add** command, as in the following example:

```
pkzip25 -add -volume test.zip *.doc
```

The volume label for the current drive will be stored.

If you wish to specify the drive, use the **volume** option followed by the drive letter with the **add** command, as in the following example:

```
pkzip25 -add -volume=a test.zip *.doc
```

## Specifying the Date of a .ZIP File

### **zipdate**

When you create a .ZIP file, PKZIP automatically applies the current date as the date of the .ZIP file. However, you can specify a different date for the .ZIP date by using the **zipdate** option with the **add** command.

PKZIP provides several methods for applying a date to a .ZIP file. The table below lists the available sub-options for applying date information to your .ZIP archives:

Sub-Option:	To use:	For example:
<b>retain</b>	The date that the .ZIP file was created.	pkzip25 -add -zipdate=retain test.zip *
<b>none</b> <b>(Default)</b>	The current date.	pkzip25 -add -zipdate=none test.zip *
<b>oldest</b>	The date of the oldest file within the .ZIP file.	pkzip25 -add -zipdate=oldest test.zip *
<b>newest</b>	The date of the newest file within the .ZIP file.	pkzip25 -add -zipdate=newest test.zip *

## Setting the Compression Level

---

When PKZIP compresses files, it uses a compression method that provides the best balance of compression and speed. If you want more compression or faster compression, you can change the default method.

PKZIP contains ten levels of compression, with 0 (zero) being no compression at the fastest speed, and 9 being the most compression at the slowest speed. For example, 1 would offer more compression than 0 (zero) but less compression than 2. However, 1 would be faster than 2, and so on.

With PKZIP, you can change the compression level in two ways. You can:

- Type an option "and" a number that represents the compression level. (This way contains one word as the option (**level**), and a number that represents the compression level.)
- Type an "option" that represents a compression level. Each option represents a compression level and is equivalent to a corresponding "number" in the first method described above (for example, the **fast** option is the same as level 2). However, this method contains only "five" levels instead of "ten".

Refer to the sections that follow for more information.

## Specifying a Compression Level by Number

### ***level***

To change the compression level, you can use the ***level*** option, followed by an equal sign, and a number that represents the compression level. The default compression level is 5 (normal), which is the best balance of compression and speed. If you don't change the compression level, PKZIP uses this level.

For example, if you wish to change the compression level to 2, you could type a command line similar to the following:

```
pkzip25 -add -level=2 test.zip *.doc
```

Since **level** is an option, it must be used in conjunction with a main command in your PKZIP command line. In this example, since you are compressing files, it is used with the **add** command.

Remember that 0 (zero) specifies no compression and therefore the fastest speed, while 9 specifies maximum compression but at the slowest speed. The higher the number, the more compression is increased at the expense of speed. As you lower that number, you get less compression, but an increase in speed. PKZIP compresses your files accordingly, depending on what level you select.

The default compression level is specified in the PKZIP Configuration Settings screen after the listed Compression item. If you wish to change the default compression level, you can do so by using the **configuration** option with the **level** option. For example, to set your default compression level to 9 (Maximum Compression), enter the following:

```
pkzip25 -config -level=9
```

For more information on changing your default settings, see *Chapter 6 - Changing Defaults Using the Configuration File* on page 107.

## Specifying a Compression Level by Option

### ***speed, fast, maximum, store, normal***

To change the compression level, you can just type a word that represents a compression level. This allows you to include a more descriptive word in your command (instead of an arbitrary number, as highlighted in the previous method). However, keep in mind that using these options offers only "five" levels of compression instead of "ten".

The available options are listed in the following table:

Option:	Level:	For example:
<b>speed</b>	For the "fastest" level of compression (level 1)	pkzip25 -add -speed test.zip *.doc
<b>fast</b>	For the "second" fastest level of compression (level 2)	pkzip25 -add -fast test.zip *.doc
<b>maximum</b>	For the highest level of compression (level 9)	pkzip25 -add -max test.zip *.doc
<b>store</b>	For "no" compression (just store files inside the .ZIP file) (level 0)	pkzip25 -add -store test.zip *.doc
<b>normal</b> <b>(Default)</b>	For normal compression (best balance of compression and speed) (level 5)	pkzip25 -add -norm test.zip *.doc  <b>Note:</b> You would only need to use this option if you changed your defaults in the Configuration file. Refer to <i>Chapter 6 - Changing Defaults Using the Configuration File</i> on page 107 for more information.

## Sorting Files Within a .ZIP File

---

### *sort*

With PKZIP, you can sort the files within a .ZIP file in several ways. If you do not change the sort order, the files are automatically sorted by the order in which they were compressed into the .ZIP file. This is called the "natural" order.

The **sort** option works with the **add**, **extract**, **test**, and **view** commands. The value you include with **sort** depends on the command you select.

Sub-Option:	To sort by:	For example:
<b>date</b>	file date	pkzip25 -add -sort=date temp.zip
<b>size</b>	original uncompressed size of the file ("length" in display)	pkzip25 -add -sort=size temp.zip
<b>extension</b>	file extension	pkzip25 -add -sort=ext temp.zip
<b>name</b>	file name (alphabetically)	pkzip25 -add -sort=name temp.zip
<b>none</b>	no sorting done	pkzip25 -view -sort=none temp.zip
<b>natural</b> <b>(same as none)</b>	the order that files exist within a .ZIP file	pkzip25 -view -sort=natural temp.zip
<b>ratio</b>	ratio of uncompressed size to compressed size	pkzip25 -view -sort=ratio temp.zip <b>Note:</b> The ratio sub-option will not work with the add command.
<b>crc</b>	CRC (Cyclic Redundancy Check) number	pkzip25 -view -sort=crc temp.zip <b>Note:</b> The crc sub-option will not work with the add command.

If you specify the **sort** option on your command line but do not specify a sub-option value, the files will be sorted by file name.

**Note:** Using the **sort** option with the **add** command only works on new .ZIP files. It does not work with a .ZIP file being updated.

## Moving Files to a .ZIP File

---

### ***move***

Normally, when you compress files, you end up with two copies of each file: the original file and the compressed file. With PKZIP, you can choose to remove the original file "after" you compress it into the .ZIP file.

If you want to move only specific files, you must compress them separately since you can only move all or none of the files that you are compressing.

To move files, use the ***move*** option with the ***add*** command, as shown below:

```
pkzip25 -add -move test.zip *.doc
```

In this example, PKZIP will compress all files that end in .doc and will remove them from their original location "after" compression.

**CAUTION:** Be careful when "moving" files from your hard drive to a .ZIP archive. If the .ZIP archive is subsequently damaged, lost, or deleted, any files contained therein will be damaged, lost, or deleted with it. Regularly back up your .ZIP archives to avoid such problems.

## Creating Self-Extracting .ZIP Files

---

### *sfx*

PKZIP allows you to create self-extracting files. In addition, you can convert existing standard .ZIP files into self-extracting files. A self-extractor is a type of .ZIP file that contains extraction code within the file itself. A self-extracting .ZIP file is an executable file with a .exe file extension (instead of .ZIP). This type of .ZIP file allows the person who is going to extract the file(s) to simply type the name of the file at the command prompt - without using the **extract** command. An external program, such as PKZIP, is not necessary to extract files from self-extracting files. For example, if the name of the self-extractor is doc.exe, you could simply type **doc.exe** at the command prompt and press ENTER to extract the files archived in that file.

Depending on whether you are using the Shareware, Registered, or Distribution Licensed version of PKZIP, you can create the following types of self-extractors:

- A native self-extractor that works with the Operating System on which PKZIP is running. (Included in the Registered, Distribution Licensed, and Shareware versions of PKZIP)
- Regular DOS: a DOS self-extractor. (Included in the Registered and Distribution Licensed versions of PKZIP)
- DOS Junior: a smaller, more compact version of the regular DOS self-extractor. This file contains less extraction options than the regular DOS self-extractor, and requires less memory to run. (Included in the Registered and Distribution Licensed versions of PKZIP)
- Windows 16-bit: a 16-bit "Windows" self-extractor that works with versions of Windows starting with 3.1 and later. This includes long file name support under Windows95. (Included in the Registered and Distribution Licensed versions of PKZIP)

- Windows 32-Bit: a 32-bit "Windows" self-extractor that works with versions of Windows95 and NT (Intel, DEC Alpha, Power PC). This includes long file name support under Windows95 and Windows NT. This self-extractor is included in the PKSFXS.DAT file described on page 80. (Available in the Distribution Licensed version of PKZIP)
- OS/2 self-extractor that works with OS/2 Operating System. (e.g. OS/2 Warp version 4). This self-extractor is included in the PKSFXS.DAT file described on page 80. (Available in the Distribution Licensed version of PKZIP)
- UNIX self-extractors that work with several different flavors of UNIX. (e.g. Sun Solaris 2.x SPARC). These self-extractors are included in the PKSFXS.DAT file described on page 80. (Available in the Distribution Licensed version of PKZIP)

To create a self-extractor, use the **sfx** option with the **add** command. If you wish to convert a standard .ZIP file to a self-extracting one, use the **sfx** option alone. The value you include with **sfx** depends on the type of file you wish to create. Subsequent sections in this chapter will more fully describe how to create PKZIP self-extracting files.

## Creating a Native Self-Extractor

To create a self-extractor native to the operating system on which you are using PKZIP, use the **sfx** option without a sub-option, as in the following example:

```
pkzip25 -add -sfx test *.doc
```

## Creating a Regular DOS Self-Extractor

To create a regular DOS self-extractor, use the **dosfull** value with the **sfx** option, as in the following example:

```
pkzip25 -add -sfx=dosfull test *.doc
```

## Creating a DOS Junior Self-Extractor

To create a DOS junior self-extractor, use the **jrds** value with the **sfx** option, as in the following example:

```
pkzip25 -add -sfx=jrds test *.doc
```

## Creating a Windows 16-Bit Self-Extractor

To create a Windows 16-bit self-extractor, use the **win16** value with the **sfx** option, as in the following example:

```
pkzip25 -add -sfx=win16 test *.doc
```

## Creating Self-Extractors with the pksfxs.dat file

The Distribution Licensed version of PKZIP includes an external data file called `pksfxs.dat`. It is *only* available and will *only* function with the Distribution Licensed version of PKZIP. The `pksfxs.dat` file allows users to create self-extractors (in addition to the extractors included with the Shareware and Registered versions of PKZIP) for several different platforms. The ***listsfxtypes*** option will display a list of the types of SFX files that can be created with PKZIP, including self-extractors contained in the `pksfxs.dat` file.

You can specify the location of the `pksfxs.dat` file on your system with the help of an environment variable called `PKSFXDATA`. For information on setting the `PKSFXDATA` environment variable, see the **Getting Started Manual**.

PKWARE will periodically update the `pksfxs.dat` file as self-extractors are developed for other operating systems and platforms. As updates are created, they will be made available to Distribution Licensed users of PKZIP for download. Contact PKWARE Technical Support for further information.

## Available Options in the Native Self-Extractor

Below is a list of PKZIP options that you may use with the native self-extractor:

- after, before, console, directories, exclude, extract, help, id, include, license, locale, mask, more, noextended, overwrite, password, permission, print, silent, sort, test, times, version, volume, warning

For example, if you wanted to recreate the directory structure stored within a self-extracting archive called test.exe, you would type the following:

```
test.exe -directories
```

Keep in mind that these options are *only* available for the native Self-Extractor and only when the Self-Extractor is being run. These options are *not* available for the DOS or Windows versions of the self-extractor.

## Differences between a Regular DOS and DOS Junior Self-Extractor

Self-extracting files (.exe) are larger than the corresponding regular .ZIP file because of the extraction code that is included. The main difference between a "regular" and a "junior" DOS self-extractor is that the junior is smaller, and contains fewer available options.

## Converting a .ZIP file to a Self-Extracting file

If you wish to convert an existing .ZIP file to a self-extracting file, use the **sfx** option in your command line. Additionally, the **sfx** option can be used to update existing self-extracting files. For example, if you wish to convert a file called test.zip into a self-extracting file called test.exe, enter the following on your command line:

```
pkzip25 -sfx test.zip
```

If you use the **sfx** option in this fashion, you will not be able to rename the file. PKZIP will take the original file name and only change the file extension from .ZIP to .exe. However, if you wish to specify a file name for the converted self-extracting file, you can use the **namesfx** option to do so. For example, if you wish to convert a file called test.zip into a self-extracting file called test123.exe, enter the following on your command line:

```
pkzip25 -sfx -namesfx=test123.exe test.zip
```

**Note:** It is important to include the .exe extension with the new file name.

## Compressing Files to Diskette (DOS)

---

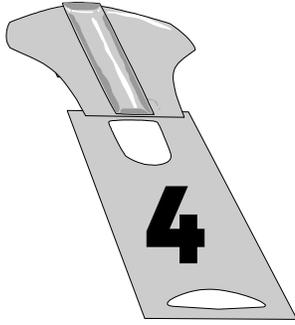
With PKZIP, you can save your .ZIP file or self-extracting file to a diskette while you create it (instead of saving it on your hard disk drive). Depending on the size of the .ZIP file, it may be necessary for PKZIP to save the file on multiple diskettes. This process is called "spanning". PKZIP automatically handles spanning by prompting you for an extra diskette (or other medium).

To compress files to diskette:

1. Insert a diskette (or other appropriate medium) into your disk drive.
2. Type your PKZIP command, and press ENTER. Make sure to specify the drive letter or path that corresponds to your destination drive. A sample command line appears below:

```
pkzip25 -add a:/test.zip *.doc
```

**Note:** Because PKZIP automatically detects removable media and therefore handles spanning automatically, it is not necessary to use an option with the add command. However, if PKZIP is unable to detect the fact that you are placing your .ZIP file on removable media, you can force spanning with the **span** option. There is a complete description of the **span** option on page 138.



# Extracting Files

## Overview:

This chapter contains information on extracting files, including all the commands and options that make data extraction more flexible.

## Sections in This Chapter:

- Introduction
- Extracting New and Existing Files
- Extracting Files Based on Some Criteria
- Extracting Files in a Specific Format
- Determining How to Handle Overwriting Files
- Retaining Directory Structure While Extracting
- Sorting Files in the Extract Directory
- Extracting Files Only for Display
- Extracting Files With a LIST File
- Authenticity Verification

## Introduction

---

PKZIP offers several options for extracting files from .ZIP archives, giving the user full control over the files stored in a PKZIP archive. This chapter contains all the information you need to use PKZIP's comprehensive extraction features.

### Conventions in This Chapter

In some of the headings in this chapter, a word appears immediately below the heading. That word represents the actual command or option you would type in your PKZIP command (and in many cases, a value or suboption in addition to the command e.g. **extract=all**). If the text under the heading is the command followed by an equal sign and another word, it means that the other word is a value (sub-option) that goes with the command.

### Default Values for Commands and Options

For each extraction task in this chapter, the command or option that represents that task contains a default value. A default value represents the action that occurs when only the name of the command or option is included in your PKZIP command. For example, the default for the **extract** command is to unzip or uncompress "all" files in an archive.

For commands and options that contain defaults, these defaults are stored in the PKZIP Configuration Options file. You can "change" defaults by editing this file with PKZIP. Refer to *Chapter 6 - Changing Defaults Using the Configuration File* on page 107 for more information.

## Extracting New and Existing Files

---

When you extract files from a .ZIP file, you can select those files you wish to extract and those you do not. If the directory into which you extract the files contains files that have the same name as those being extracted, you have to decide if you want to overwrite those files.

PKZIP provides several methods that allow you to determine which files to extract. You can extract:

- All files in a .ZIP file.
- Files that are "newer" than files with the same name in the extract directory "and" files that do not appear in the extract directory.
- Only files that are "newer" than the files with the same name in the extract directory.

Refer to the sections that follow for more information.

## Extracting All Files From a .ZIP File

### ***extract***

To extract all files from a .ZIP file, type ***pkzip25 -extract*** and the name of your .ZIP file, as shown below:

```
pkzip25 -extract test.zip
```

In this example, all files in the .ZIP archive will be extracted into the current directory.

## Overriding Default Files to Extract

### ***extract=all***

With PKZIP, you can set up default files to extract. For example, if you always (or usually) extract .doc files, you can set up PKZIP to automatically extract those files, without specifying those files in your command. Refer to *Chapter 6 - Changing Defaults Using the Configuration File* on page 107 for more information.

To override defaults to extract all files from a .ZIP file, you can use the ***all*** value with the ***extract*** command, as shown below:

```
pkzip25 -extract=all test.zip
```

In this command, you do not have to specify the files to extract because you are extracting all files. In addition, the default that is set up in the Configuration file still applies. The ***all*** value only overrides the default for this instance only.

## Extracting Newer Versions of Existing Files and New Files

### ***extract=update***

PKZIP allows you to extract files contained in a .ZIP archive that are newer versions of files than those that already exist in the extract directory. For example, if you have a file called apples.doc in the extract directory, but a newer version in the .ZIP file, you can overwrite the older version with the newer version using the ***update*** sub-option.

The following example uses the ***update*** sub-option with the ***extract*** command:

```
pkzip25 -extract=update test.zip
```

When the **update** sub-option is used, the files specified for extraction in the .ZIP file will be compared against the files already present in the extraction directory. If PKZIP determines that the file(s) to be extracted are older than the same file(s) that already exist in the extraction directory, PKZIP will not extract the file(s). Additionally, any file(s) that exist in the .ZIP file but not in the extract directory are automatically extracted to the extract directory.

## Extracting Only Newer Versions of Files

### ***extract=freshen***

To extract only those files in destination directory that have changed, use the **freshen** sub-option with the **extract** command, as shown below:

```
pkzip25 -extract=freshen test.zip
```

When you use **freshen** sub-option with the **extract** option, only those files in the .ZIP file which already exist in the extract directory, and that have a file modification date newer than the files already in the extract directory, will be extracted from the .ZIP file.

## Extracting Files Based on Some Criteria

---

### ***after, before, include, exclude***

With PKZIP, you can extract files based on certain criteria. You can extract:

- Files that are newer than a specified date.
- Files that are older than a specified date.
- All files of a specific pattern.
- All files "except" those specified.

Refer to the sections that follow for more information.

## Extracting Files Newer Than a Specified Date

### ***after***

With PKZIP, you can choose to add or extract only those files that are newer or equal to a specified date. To do this, use the ***after*** option, followed by an equal sign and the date, as shown below:

```
pkzip25 -extract -after=062495 test.zip
```

In this example, only files that contain a date newer than or equal to June 24, 1995 will be extracted.

With PKZIP, you can enter dates in the following formats:

- mmddyy
- mmddyyyy

This order in which you type the month, day, and year depends on your ***locale*** setting. For more information on the ***locale*** setting, see *Chapter 7 - Command Characteristics* on page 114.

## Extracting Files Older Than a Specified Date

### ***before***

With PKZIP, you can choose to add or extract only those files that are older than a specified date. To do this, use the ***before*** option, followed by an equal sign and the date, as shown below:

```
pkzip25 -extract -before=062495 test.zip
```

In this example, only files that contain a date older than June 24, 1995 will be extracted.

With PKZIP, you can enter dates in the following formats:

- mmddyy
- mmddyyyy

This order in which you type the month, day, and year depends on your **locale** setting. For more information on the **locale** setting, see *Chapter 7-Command Characteristics* on page 114.

## Including Files That Match a Pattern

### ***include***

PKZIP allows you to add or extract files that match a specific pattern, for example, all files that end in the .doc extension.

To extract files that match a pattern, simply include the pattern after the **extract** command, as in the following example:

```
pkzip25 -extract test.zip *.doc
```

You have the option of specifying a default file pattern setting in your PKZIP Configuration file. If, for example, you want to automatically include all files with the extension of .doc in PKZIP compress and extract operations, enter the following:

```
pkzip25 -config -include=*.doc
```

When you use the **include** option with the **configuration** command, PKZIP prompts you to configure the **include** default for add *and/or* extract operations. The .doc file pattern will henceforth be included by default in compress *and/or* extract operations.

If you set up a default pattern in your Configuration file, but want to specify a different pattern for this instance only (for example, extract all files that end in .txt), use the **include** option, followed by an equal sign, then the file pattern, as in the example below:

```
pkzip25 -extract -include=*.txt test.zip
```

If you don't set a default for **include** in the Configuration file, you don't have to specify the **include** option in your command, only the file pattern. However, if you specify both an **include** value as well as a file pattern, both are used in the file extraction process.

## Excluding Files from Being Extracted

### **exclude**

When you extract files, you might want to exclude specific files from being extracted, for example, all files that end in .bmp. To exclude files, use the **exclude** option with the **extract** command, as shown below:

```
pkzip25 -extract -exclude=*.bmp test.zip
```

In the example above, all files except those that end in the .bmp extension will be extracted.

You have the option of specifying a default file pattern setting in your PKZIP Configuration file. If, for example, you want to automatically exclude all files with the extension of .doc in PKZIP compress and extract operations, enter the following:

```
pkzip25 -config -exclude=*.doc
```

When you use the **exclude** option with the **configuration** command, PKZIP prompts you to configure the exclude default for add *and/or* extract operations. The .doc file pattern will henceforth be excluded by default in compress and/or extract operations.

## Extracting Files in a Specific Format (OS/2)

---

### **nametype**

The **nametype** option allows you to extract files in a specific format such as Long File Name or 8+3. You may specify how you wish files to be extracted by using the **nametype** option along with one of the sub-options listed in the table that follows:

Sub-Option:	Description:	For example:
<b>auto</b>	PKZIP will attempt to Auto-Detect the file system where the file(s) will be extracted. File(s) will be extracted accordingly.	pkzip25 -extract -nametype=auto test.zip /temp
<b>short</b>	File(s) will be extracted in the short, or 8+3 naming convention.	pkzip25 -extract -nametype=short test.zip /temp
<b>long</b>	File(s) will be extracted in the same format as they were in when originally added to the .ZIP file. This includes files in the Long File Name format.	pkzip25 -extract -nametype=long test.zip /temp

Keep in mind that PKZIP may not be able to extract files in the specified **nametype** format. The extraction format is greatly dependent on such things as the originating or destination PC's file system. If, for example, the originating PC's file system supports Long File Names but the destination PC's file system does not, you may get an error or the file name may be automatically truncated, regardless if you use the **long** sub-option.

**Note:** If the **nametype** option is not specified on the command line, PKZIP will attempt to auto-detect the file system by default. If, however, you *do* specify the **nametype** option on your command line, but do not specify a sub-option, files will be extracted with short file names

## Determining How to Handle Overwriting Files

---

### **overwrite**

When you extract files from a .ZIP file, you might be extracting those files into a directory that already contains a copy (or version) of those files. For more control in determining whether or not to overwrite duplicate files with the corresponding "extracted" files, use the **overwrite** option with the **extract** command. PKZIP provides three methods for using the **overwrite** option.

The available sub-options are listed in the table that follows.

Sub-Option:	Description:	For example:
<b>prompt</b>	PKZIP will prompt you to overwrite a duplicate file before proceeding.	<code>pkzip25 -extract -overwrite=prompt test.zip *.bmp</code>
<b>never</b>	PKZIP will not overwrite any duplicate files.	<code>pkzip25 -extract -overwrite=never test.zip *.bmp</code>
<b>all</b>	PKZIP will overwrite all duplicate files. You will not be prompted.	<code>pkzip25 -extract -overwrite=all test.zip *.bmp</code>

**Note:** If you use the **extract** option *alone* without the **overwrite** option, you will be prompted to overwrite duplicate files. If, however, you use the **extract** option as well as the **overwrite** option but do *not* specify a sub-option, PKZIP will automatically overwrite all files without prompting you.

## Retaining Directory Structure while Extracting

---

### *directories*

If you stored directory path information within a .ZIP file, you can re-create those directory paths when you extract the files. For example, if you compressed a file called `apples.doc` in the `temp/fruit` directory, and you stored `temp/fruit` you can re-create `temp/fruit` in the location in which you extract the files.

To re-create directories, use the **directories** option with the **extract** command, as in the following example:

```
pkzip25 -extract -directories test.zip *
```

When you use this command, all directories that were stored in the .ZIP file will be retained during extraction. The directory path stored is appended to the directory in which you extract the files. For example, if your extract directory is /doc, and a directory path stored with the files is temp/fruit, the files would now be extracted to /doc/temp/fruit.

## Sorting Files in the Extract Directory

---

### **sort**

PKZIP allows you to specify the sort order of files that are compressed in a .ZIP file or extracted into a destination directory. For example, if you wish to extract files in a specified sort order (by date), you would type the following and press ENTER:

```
pkzip25 -extract -sort=date test.zip
```

In this example, all files that exist in the test.zip file are extracted into the current directory sorted in ascending order by date. For further information on the available sort options, see page 137, the on-line help, or the *Sorting Files Within a .ZIP File* section on page 75.

## Extracting Files Only for Display

---

### **console**

PKZIP gives you the option of displaying specific files contained in a .ZIP file to your computer monitor. For example, if you wish to view the contents of all of the .txt files contained in a .ZIP file, type the following and press ENTER:

```
pkzip25 -console test.zip *.txt
```

In this example, all files with a .txt extension that exist in the test.zip are displayed on the monitor. Since many .ZIP files contain an information document (e.g. readme.txt), the **console** option is a good way to determine the contents of a .ZIP file without requiring you to extract a file or file(s) to your hard drive.

## Extracting Files with a LIST File

---

The file format of a LIST file used to extract or exclude certain files is somewhat different than the format used to include files. When compressing files, the LIST file needs to be in a format that the operating system can understand. For example, in DOS based command lines, it may be necessary to specify a drive letter when compressing files. By contrast, when extracting or excluding files, a drive letter is not necessary and therefore cannot be used in LIST files.

A LIST file may contain wild card specifications (\*,?) as well as exact file names and paths. An example of a LIST file used in extract operations follows:

```
*.exe
*.doc
temp/readme.doc
temp/?????.*
text/news.txt
```

Using the @ character in your command line, you can point to the LIST file. We'll assume that the LIST file is called lst.txt and is located in your current directory. An example of such a command line follows:

```
pkzip25 -extract test.zip @lst.txt
```

In the example above, PKZIP extracts files from test.zip using file information it retrieves from a file called lst.txt, located in the current directory. It is important that the file format contained in the LIST file matches the format of the files in the test.zip file. For example, if test.zip contains no path information, specifying **text/news.txt** in your LIST file will *not* extract the file **news.txt** from the test.zip file. See the *Compressing Files with a LIST File* section on page 54 for more information on LIST files. For information on viewing the files as they appear in the .ZIP file, refer to the *Viewing the Contents of a .ZIP File* section on page 98.

## Authenticity Verification

---

This version of PKZIP allows you to authenticate .ZIP files created with Authenticity Verification (AV) information. You can *not* embed AV information into archives created with this version of PKZIP. However, you can test files created with other versions of PKZIP for authenticity. For example, if you created a .ZIP archive in PKZIP 2.04g for DOS with AV information embedded in that archive, you could then use PKZIP to verify that the files in that archive have not been tampered with or modified. PKZIP automatically checks for AV information when extracting files.

An example of a successful extraction of files that were originally compressed with AV information follows:

```
PKZIP(R) Version 2.50 FAST! Compression Utility
Copyright 1989-1997 PKWARE Inc. All Rights Reserved. Registered Version
PKZIP Reg. U.S. Pat. and Tm. Off. Patent No. 5,051,745

Extracting files from .ZIP: test.zip

    Inflating: file1.dat  -AV
    Inflating: file2.dat  -AV

Authentic files verified!
# XYZ123  Vandelay Industries
```





## Performing Miscellaneous Tasks on .ZIP Files

### Overview:

This chapter contains information on additional tasks that you can perform on .ZIP files. The options discussed in this chapter are not as commonly used as the options discussed in previous chapters, but useful nonetheless.

### Sections in This Chapter:

- Introduction
- Viewing the Contents of a .ZIP File
- Printing the Contents of a .ZIP File
- Testing the Integrity of a .ZIP File
- Previewing Command and Option Operations
- Fixing a Corrupt .ZIP File
- Create a Temporary .ZIP File On Alternate Drive
- Suppressing Screen Output

## Introduction

---

PKZIP offers several miscellaneous tasks you can perform on .ZIP files. You can:

- View the contents of a .ZIP file.
- Print files contained in a .ZIP file.
- Test the integrity of a .ZIP file.
- Previewing command and option operations.
- Fix a corrupt .ZIP file.

Refer to the sections that follow for more information.

## Viewing the Contents of a .ZIP File

---

### ***view***

PKZIP allows you to view the contents of a .ZIP file, without performing any action on that .ZIP file (for example, compress or extract). To view a .ZIP file, use the ***view*** option with PKZIP, as in the following example:

```
pkzip25 -view test.zip
```

When you type this command, information similar to the following appears:

```
PKZIP(R) Version 2.50 FAST! Compression Utility
Copyright 1989-1997 PKWARE Inc. All Rights Reserved. Registered Version
PKZIP Reg. U.S. Pat. and Tm. Off. Patent No. 5,051,745

Viewing .ZIP: test.zip

Length Method Size Ratio Date Time CRC-32 Mode Name
-----
 3746 DeflatN 1425 62.0% 03/11/1997 4:16p e4da66f6 0666 readme.txt
 1986 DeflatN 812 59.2% 03/11/1997 8:51a 7bcc0521 0666 whatsnew.txt
-----
 5732 2237 61.0% 2
```

**Note:** The above **view** list was generated from a UNIX command line. In a DOS **view** listing, the "Mode" column would be replaced by an attributes "Attr" column.

PKZIP also provides two additional methods for displaying information from a .ZIP file. Specify the desired method as a value in addition to the **view** option. These methods include:

- brief - a compact, less informative view of the .ZIP file.
- detail - more information than the default view.

## Displaying a Brief View of a .ZIP File

To display a more compact (brief) view of a .ZIP file, use the **brief** value with the **view** option, as in the following example:

```
pkzip25 -view=brief test.zip
```

When you press ENTER, information similar to the following appears:

```
PKZIP(R) Version 2.50 FAST! Compression Utility
Copyright 1989-1997 PKWARE Inc. All Rights Reserved. Registered Version
PKZIP Reg. U.S. Pat. and Tm. Off. Patent No. 5,051,745

Viewing .ZIP: test.zip

Length Method Size Ratio Date Time Name
-----
 3746 DeflatN 1425 62.0% 03/11/1997 4:16p readme.txt
 1986 DeflatN 812 59.2% 03/11/1997 8:51a whatsnew.txt
-----
 5732 2237 61.0% 2
```

## Displaying a Detailed View of the .ZIP File

To display a more detailed view of a .ZIP file, use the **details** value with the **view** option, as in the following example:

```
pkzip25 -view=details test.zip
```

When you press ENTER, information similar to the following appears:

```
PKZIP(R) Version 2.50 FAST! Compression Utility
Copyright 1989-1997 PKWARE Inc. All Rights Reserved. Registered Version
PKZIP Reg. U.S. Pat. and Tm. Off. Patent No. 5,051,745

Viewing .ZIP: test.zip

      FileName: readme.txt
      FileType: binary
      Mode: 0666
      Date and Time: Mar 11,1997  4:16:28p
      Compression Method: DeflatN
      Compressed Size: 1425
      Uncompressed Size: 3746
      Compression: 62.0% - 3.043 bits/byte
      32 bit CRC value: e4da66f6
      Version created by: PKZIP: 2.5
      Needed to extract: PKZIP: 2.0 or later

      FileName: whatsnew.txt
      FileType: binary
      Mode: 0666
      Date and Time: Mar 11,1997  8:51:54a
      Compression Method: DeflatN
      Compressed Size: 812
      Uncompressed Size: 1986
      Compression: 59.2% - 3.270 bits/byte
      32 bit CRC value: 7bcc0521
      Version created by: PKZIP: 2.5
      Needed to extract: PKZIP: 2.0 or later

-----

      Total Files: 2
      Compressed Size: 2237
      Uncompressed Size: 5732
      Compression: 61.0% - 3.122 bits/byte
```

**Note:** The above **view** list was generated from a UNIX command line. In a DOS **view** listing the "Mode" column would be replaced by an attributes "Attr" column.

## Printing the Contents of a .ZIP File (DOS, OS/2)

---

### *print*

PKZIP gives you the option of printing files contained in a .ZIP file to a selected printer. For example, if you wish to print all of the .txt files contained in a .ZIP file, type the following:

```
pkzip25 -print=lpt1 test.zip *.txt
```

When you press ENTER, information similar to the following will appear:

```
PKZIP(R) Version 2.50 FAST! Compression Utility
Copyright 1989-1997 PKWARE Inc. All Rights Reserved. Registered Version
PKZIP Reg. U.S. Pat. and Tm. Off. Patent No. 5,051,745

Extracting files from .ZIP: test.zip

Inflating: readme.txt <to LPT1>
Inflating: whatsnew.txt <to LPT1>
```

In this example, all files with a .txt extension that exist in the test.zip are printed to the LPT1 printer. If you do not specify a print device, the 'default' printer is used. Since many .ZIP files contain an information document (e.g. readme.txt), the *print* option is a good way to determine the contents of a .ZIP file without requiring you to extract a file or file(s) to your hard drive.

## Testing the Integrity of a .ZIP File

---

### *test*

PKZIP allows you to test .ZIP files to verify that they are not damaged. Before storing an important .ZIP file or sending it to another person, it is a good idea to test it first. For example, if you wish to test the contents of test.zip, type the following:

```
pkzip25 -test test.zip
```

When you press ENTER, information similar to the following will appear:

```
PKZIP(R) Version 2.50 FAST! Compression Utility
Copyright 1989-1997 PKWARE Inc. All Rights Reserved. Registered Version
PKZIP Reg. U.S. Pat. and Tm. Off. Patent No. 5,051,745

Testing files from .ZIP: test.zip

Testing: readme.txt OK
Testing: whatsnew.txt OK
```

As each file is tested, an OK is displayed next to the name. If, for some reason, the archive has been damaged, use the **fix** option described on page 103 to repair the .ZIP file.

## Previewing Command and Option Operations

---

### *preview*

PKZIP allows you to preview the results of a set of commands and options. The commands and options specified will be completed and the resulting output will display, but no changes will be made that result in creating a new .ZIP file or in modifying an existing .ZIP file. For example, if you wish to preview an add operation without actually creating or modifying any files, enter the following:

```
pkzip25 -add -preview test.zip *.txt
```

When you press ENTER, information similar to the following appears on your console:

```
PKZIP(R) Version 2.50 FAST! Compression Utility
Copyright 1989-1997 PKWARE Inc. All Rights Reserved. Registered Version
PKZIP Reg. U.S. Pat. and Tm. Off. Patent No. 5,051,745

◆ Using Preview Option

Creating .ZIP: test.zip
Adding File: readme.txt Deflating (62.0%), done.
Adding File: whatsnew.txt Deflating (59.2%), done.

The compressed .ZIP file size would be: 2237 bytes
```

The information, including the size of the resulting .ZIP file, is displayed. However, PKZIP has not actually modified any of your files. The **preview** option will work with the **add**, **delete**, **header**, **sfx** and **comment** commands.

## Fixing a Corrupt .ZIP File

---

### ***fix***

In the event that a .ZIP file becomes damaged, you may find that it is not possible to extract or perform other PKZIP operations on the contents of the file. The ***fix*** option in PKZIP will attempt to repair damaged .ZIP archives.

For example, if you have determined that test.zip is damaged, type the following to attempt to fix it:

```
pkzip25 -fix test.zip
```

When you press ENTER, information similar to the following appears on your console:

```
PKZIP(R) Version 2.50 FAST! Compression Utility
Copyright 1989-1997 PKWARE Inc. All Rights Reserved. Registered Version
PKZIP Reg. U.S. Pat. and Tm. Off. Patent No. 5,051,745

Enter a new .ZIP file name (pkfixed): test1.zip
Running PKZipFix utility.

Scanning .ZIP file:      test.zip
Building new directory.
Writing new .ZIP file:  test1.zip

Recovered 2 files.
```

Please note that when you enter the ***fix*** option, PKZIP will prompt you to enter a new .ZIP file name. In the above example, test1.zip was entered. If you do not enter a file name, the name "pkfixed.zip" will be used. PKZIP scans the original file, attempts to repair the archive, and saves the updated file with the file name provided. The original, damaged file is not updated. The ***fix*** option will not repair all damaged .ZIP files. Depending on the degree of damage to the data within a .ZIP file, you may not be able to recover your files from that archive.

## Create a Temporary .ZIP File on a Alternate Drive

---

### *temp*

Every time you update a .ZIP file, PKZIP creates a temporary work file. Before modifying the original file, PKZIP performs all of its compression and extraction operations on the temporary work file. When the modifications to the .ZIP file are successfully completed, the original .ZIP file is replaced with the updated file (temporary work file). This means you must have as much additional disk space available as was used by the original .ZIP file. For example, if you have an existing .ZIP file of 500K and you are adding another file to it that is 10K compressed, you need additional workspace of at least 510K during the update process.

The *temp* option allows you to create the temporary .ZIP file on a drive other than the one on which the original .ZIP file resides. This allows you to update large .ZIP files when space is limited, such as a large .ZIP file on a floppy disk. Furthermore, by setting this temporary drive to point to a RAM drive, you can speed up the operation of PKZIP.

Immediately following the *temp* option, place the drive and/or path you wish to use for the temporary work file as in one of the following examples:

➤ UNIX based command line:

```
pkzip25 -add -temp=/usr/tmp test.zip readme.doc
```

➤ DOS based command line:

```
pkzip25 -add -temp=z:/public test.zip readme.doc
```

**Note:** It is necessary to specify a path in addition to the drive letter only if you are in a situation where disk space or access is being limited by subdirectory, such as on a local area network.

## Suppressing Screen Output

---

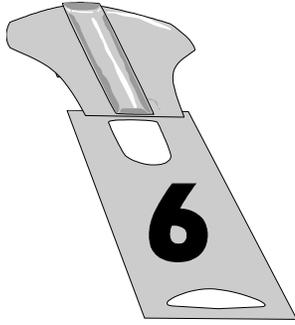
### ***silent***

The ***silent*** option suppresses screen output when compressing or extracting. This option can be used to compress and extract files as part of .BAT, .CMD, or shell script files for background processing. Messages that normally appear when compressing or extracting will not be displayed. Keep in mind that errors and warnings are displayed whether the silent option is specified or not, however, prompts for other PKZIP operations are *not* displayed (e.g. Password, Overwrite, Insert Disk). You may wish to use the ***silent*** option with the ***nofix*** option, which suppresses the "attempt to fix" prompt if PKZIP encounters errors in a .ZIP file. An example command line using these options follows:

```
pkzip25 -add -silent -nofix test.zip *.doc
```

In this example, PKZIP would add all files in the current directory with a .doc extension to a .ZIP archive called test.zip. All screen output would be suppressed. The "attempt to fix" prompt would be suppressed as well, if PKZIP encounters any errors in the test.zip file. For more information on the ***silent*** and ***nofix*** options, refer to *Appendix A* on page 119.





# Changing Defaults Using the Configuration File

## **Overview:**

This chapter contains information on changing default values in the PKZIP Configuration Settings File.

## **Sections in This Chapter:**

- Introduction
- Viewing the Configuration File
- Changing a Default Value
- Resetting Original Defaults

## Introduction

---

Most of the PKZIP commands and options contain "default" values. Not all commands and options contain more than one value. However, for those that do, you can change the default value.

Default values for PKZIP commands and options are stored in the Configuration file. To change a default, you edit this file with PKZIP.

For a quick look at the contents of this Configuration file, refer to the next section, *Viewing the Configuration File*.

To learn how to change a default, refer to the *Changing a Default Value* section on page 110.

## Viewing the Configuration File

---

Before you change a default, you might want to take a look at the contents of the Configuration file.

To access the Configuration file at your command prompt, type the following and press ENTER:

```
pkzip25 -config
```

➤ If you are using the UNIX based version of PKZIP, the following screen will appear:

```

PKZIP Configuration Settings

204          = Disabled
Add          = All
After Date   = None
Before Date  = None
Comment      = No files
Compression  = Level 5 - Default
Directory    = Disabled
Extract      = All
Header       = Disabled
List Char    = @
Locale
  Time       = 12-Hour
  Date       = MMDDYY
Move Files   = Disabled
NoExtended   = Disabled
NoFix        = Disabled
NoZipExten   = Disabled

File Permissions = Not Preserved
File Access Time = Not Preserved
File Modify Time = Not Preserved
File Types      = Regular
Follow Link Types = None

Exclude files  = None
Include files  = None

Option Char = -
Overwrite    = Prompt
Password     = Disabled
Path         = None
Recurse      = Disabled
Sfx Type     = SOL2X_SPC_C250
Shortname    = DOS
Silent       = Disabled
Sort
  Add        = Natural
  Extract    = Natural
  View       = Natural
  Test       = All
View Option  = Normal
Warn. Pause  = Disabled
Zip Date     = Disabled

File Group ID = Not Preserved
File User ID  = Not Preserved

```

**Note:** PKZIP will look in your start-up file (e.g. .login or .profile) for the PKZIPCFG environment variable, which specifies a search path to your configuration file (PKZIP.CFG). If the PKZIPCFG environment variable is not set, PKZIP will look in the current working directory for the PKZIP.CFG file and attempt to edit it. If there is no PKZIP.CFG file in the current working directory, PKZIP will create a PKZIP.CFG file (with default settings) in the current working directory. For more information on setting the PKZIPCFG environment variable, see the **Getting Started Manual**.

➤ If you are using the DOS based version of PKZIP, the following screen will appear:

```

PKZIP Configuration Settings

204          = Disabled
Add          = All
After Date   = None
Attributes   = Read-Only
Before Date  = None
Comment      = No files
Compression  = Level 5 - Default
Directory    = Disabled
Extract      = All
Header       = Disabled
List Char    = @
Locale
  Time       = 12-Hour
  Date       = MMDDYY

Mask
  Compress   = None
  Extract    = Read-Only, Hidden, System
More
  Compress   = Disabled
  Extract    = Disabled
  View       = Disabled
Move Files   = Disabled
NameType     = Auto
Option Char  = -
Overwrite    = Prompt
Password     = Disabled
NoExtended   = Disabled
NoFix        = Disabled

More - Press Enter or Space for next screen, Esc to abort

```

To view more of this file, press ENTER.

When you press ENTER, the following information appears:

```
PKZIP Configuration Settings
NoZipExten = Disabled
Path       = None
Recurse    = Disabled
Sfx Type   = WIN32_X86_C250
Shortname  =
Silent     = Disabled
Sort
  Compress = Natural
  Extract  = Natural
  View     = Natural

Test       = All
View Option = Normal
Volume
Compress   = Disabled
Extract    = Disabled
warn. Pause = Disabled
Zip Date   = Disabled

Exclude Files
Compress   = None
Extract    = None
Include Files
Compress   = None
Extract    = None
```

In this file, the command/options are to the "left" of the equal sign; the default values are to the "right" of the equal sign. Words that are indented under a command or option represent different tasks (for example, compress or extract). If a command or option can be used for two different tasks (for example, compress and extract), you can set a different default for each task.

For information on changing defaults, refer to next section, *Changing a Default Value*.

## Changing a Default Value

---

Changing a default value is similar to typing any other command using PKZIP. The only difference is that you are not compressing or extracting; you are setting a default, for example, automatically compress "only" files that end in .doc. Some of these options are configurable for both compression and extraction. For example, you can specify a default value for both compress and extract operations in such options as **exclude** and **include**.

To change a default value for a command/option:

1. Type **pkzip25 -config** and the command/option followed by an equal sign and finally the sub-option value you want to set as the default. For example, to change the default for the **add** command to **update** (instead of the original default, **all**), you would type the following:

```
pkzip25 -config -add=update
```

When you change a default value, the updated Configuration file is displayed.

**Note:** You will find a listing of configurable values for PKZIP's commands and options in *Appendix A - PKZIP Options: A Quick Reference* on page 119. Refer to the "Name Description" column to determine which values are configurable. Refer to the "Value(s)" column for a listing of the specific values.

## Resetting Original Defaults

---

For some commands and options, you might want to change defaults for a few instances, but eventually switch back to all of the original defaults. PKZIP allows you to do this using the **default** option.

You can reset:

- All defaults.
- Individual defaults.

## Resetting All Defaults

To reset default values for all commands and options (that contain default values), type the following and press ENTER:

```
pkzip25 -default
```

## Resetting Individual Defaults

PKZIP allows you to reset a default value for an individual command or option. You do not have to reset all of the commands/options at once. For example, if you wish to reset the **add** value back to its default setting without resetting any other Configuration values that you may have modified, type the following and press ENTER:

```
pkzip25 -config --add
```

The minus sign in front of the command tells the Configuration file to set the value for that particular command back to the default value. In the case of the "-" character preceding the **add** value in the example above, it changes the "Update" value we set in a previous example back to "ALL".



## Command Characteristics

### Overview:

This chapter contains information on performing tasks that pertain to the "conventions" or syntax of PKZIP commands and options. These tasks are more technical, and require at least rudimentary knowledge of the operating system.

### Sections in This Chapter:

- Introduction
- Changing Environment Variables for Dates and Times
- Changing the LIST Character for List Files
- Changing the Command/Option Character

## Introduction

---

Up until this point, we've discussed PKZIP and its commands and options in very specific syntactical terms. You may find that in certain circumstances it becomes beneficial or necessary to change the syntax of PKZIP's commands and/or options. Please keep in mind that these modifications are only useful in very specific situations and should only be made by those who are very familiar with the operating system.

## Changing Date and Time Environment Variables

---

**Note:** This feature is *only* necessary to accommodate special formats for entering dates or times. If the **mmddy** date format and/or the **hh:mm** time format is acceptable, you do not have to use this option.

### ***locale (OS/2)***

By default, PKZIP uses a 12-hour time format and MMDDYY date format when archiving files. For some systems, it may be necessary to change the defined system "locale" environment variable that defines valid formats for entering dates. With PKZIP, you can change this variable in the Configuration file or while compressing files.

To change this variable, use the ***locale*** option, followed by the country. For example, to specify Germany as the default locale variable in the Configuration file, type the following and press ENTER:

```
pkzip25 -config -locale=germany
```

You may also specify a locale variable while creating or updating a .ZIP file. The following is an example of such a command:

```
pkzip25 -add -locale=germany test.zip *.doc
```

PKZIP can also be instructed to check for and use the environment variable defined by the operating system. Specify the **environment** sub-option on your command line to use the defined environment variable. For example:

```
pkzip25 -add -locale=environment test.zip *.doc
```

Not specifying a sub-option with the **locale** option is the same as specifying the **environment** sub-option.

## ***locale (UNIX, DOS)***

By default, PKZIP uses the 12-hour time format and MMDDYY date format when archiving files. The **locale** option allows you to use the system defined time and date format settings.

To set the default PKZIP time and date settings in the Configuration file to match your system time and date settings, type the following and press ENTER:

```
pkzip25 -config -locale
```

You may also use the **locale** option while creating or updating a .ZIP file. The following is an example of such a command:

```
pkzip25 -add -locale test.zip *.doc
```

In this example, PKZIP will use the system-defined settings regardless of the time and date settings defined in the Configuration file.

## Changing the LIST Character for LIST Files

---

### *listchar*

PKZIP allows you to specify an ASCII file as a source list of the files to be archived. By default, you specify this ASCII file by pointing to it with the "@" character in your command line. However, if you have files that begin with an "@", you may experience problems when trying to add these files to a .ZIP archive. Fortunately, PKZIP allows you to change the default list character to avoid such problems. This is accomplished using the *listchar* option. For example, if you wish to define the "+" character in place of the "@" as your default list character, type the following and press ENTER:

```
pkzip25 -config -listchar=+
```

If you wish to specify an alternate list character on the command line itself, could type a command line similar to the following and press ENTER:

```
pkzip25 -add -listchar=+ test.zip +file1.txt
```

When used as a command line option, the *listchar* option only applies to the options that follow it on that particular command line. In our example the *listchar* option allows you to add files that begin with a "+" character (e.g. +file1.txt). For more information on using LIST files with PKZIP see the *Compressing Files with a LIST File* section on page 54 and the *Extracting Files with a LIST File* section on page 94.

**Note:** Avoid using metacharacters as list characters. Metacharacters have a special significance to the shell and as such their usage may cause unpredictable results. This would include the following characters:

```
; , & ( ) | < > NEWLINE SPACE TAB
```

## Changing the Command/Option Character

---

### *optionchar*

All commands and options illustrated thus far have been preceded by a "-" character. The "-" character is the default option character used on the command line to indicate an option or command. PKZIP allows you to change the default option character to some other specified ASCII character. By default, the option is set to the "-" character. However, if you have file names that, for example, begin with an "-", you may experience problems when trying to add these files to a .ZIP archive. Fortunately, PKZIP allows you to change the default option character to avoid such problems. This is accomplished using the *optionchar* option. For example, if you wish change the default option character from "-" to "+", type the following:

```
pkzip25 -config -optionchar=+
```

You would then have the option of using the "+" character to indicate a command or option in your command line. Additionally, the "/" character can always be used to indicate a command or option in a DOS based command line. If you wish to specify an alternate option character on the command line itself, type the following and press ENTER:

```
pkzip25 -opt=+ +add save.zip *.doc
```

When used as a command line option, the *optionchar* option only applies to the options that follow it on that particular command line. In our example, the *optionchar* option allows you to use the "+" character with the *add* option.

**Note:** Avoid using metacharacters as option characters. Metacharacters have a special significance to the shell and as such their usage may cause unpredictable results. This would include the following characters:

```
; , & ( ) | < > NEWLINE SPACE TAB
```





## Appendix A

### PKZIP Options: A Quick Reference

This appendix contains reference information on every PKZIP command and option. For each command/option, the following information is provided:

Category:	Represents:
<b>Name/Description</b>	The full name of the command/option and a brief description of what that command/option does.  If the command/option can be configured (defaulted) in the Configuration file, the word "configurable" appears after the description.
<b>Value(s)</b>	The value(s) associated with this command/option, including the "default" value for each.  If a command/option does not have an associated value, the phrase "no suboptions" appears.
<b>Example usage</b>	An example of how to include this command/option in your PKZIP command line, including possible abbreviations. For most options, you can abbreviate the command/option. These abbreviations are illustrated in the examples used in this appendix.
<b>Used with</b>	This command can be used for compression, extraction, viewing, testing, a combination, or as a stand-alone (none of the above).

Information on each command/option follows:

Name/Description:	Value(s):	Example usage:	Used with:
<p><b>204</b></p> <p>turns on PKZIP for DOS 204g compatibility</p> <p>configurable</p>	<p>no suboptions</p> <p>-----</p> <p>no default value</p>	<p>pkzip25 -add -204 save.zip *</p>	<p>add</p>
<p><b>add</b></p> <p>add files to a .ZIP file</p> <p>configurable</p>	<p><b>all</b> - compress new files and all existing files</p> <p><b>freshen</b> – compress only files that exist in the .ZIP and that have changed</p> <p><b>update</b> – compress new and update existing files</p> <p><b>incremental</b> - compress only files that have the archive attribute on and subsequently turns off the archive attribute (<b>DOS, OS/2</b>)</p> <p><b>-incremental</b> - compress only files that have the archive attribute on and does not turn off the archive attribute (<b>DOS, OS/2</b>)</p> <p>-----</p> <p>default = <b>all</b></p>	<p>pkzip25 -add save.zip *.doc</p> <p>pkzip25 -add=freshen save.zip *.doc</p> <p>pkzip25 -add=increm save.zip *.doc</p> <p>pkzip25 -add=-increm save.zip *.doc</p>	<p>stand-alone</p>
<p><b>after</b></p> <p>process files that are newer than, or equal to, a specified date</p> <p>configurable</p>	<p>any date in format specified in Country-Settings or the locale option</p> <p>e.g. the US date format is:</p> <p>mmddyy</p> <p>or</p> <p>mmddyyyy</p> <p>-----</p> <p>no default value</p>	<p>for compression:</p> <p>pkzip25 -add -aft=091595 save.zip *.doc</p> <p>for extraction:</p> <p>pkzip25 -ext -after=091595 save.zip *.doc</p>	<p>add, extract, delete, test, view</p>





Name/Description:	Value(s):	Example usage:	Used with:
<p><b>attributes</b></p> <p>stores files with the specified file attribute information in the .ZIP file</p> <p>configurable</p> <p><b>(DOS, OS/2)</b></p>	<p><b>hidden</b> - compress hidden files</p> <p><b>system</b> - compress system files</p> <p><b>readonly</b> - compress read-only files</p> <p><b>all</b> - compress all types of files</p> <p><b>none</b> - don't compress files that have hidden, system, or read-only attributes; overrides the default attributes setting in configuration file</p> <p>-----</p> <p>default = <b>readonly</b></p> <p>default if used on command line without a sub-option = <b>all</b></p>	<p>pkzip25 -add -attr=system,hidden save.zip *</p>	<p>add</p>
<p><b>before</b></p> <p>process files that are older than a specified date</p> <p>configurable</p>	<p>any date in format specified in Country-Settings or the locale option</p> <p>e.g. the US date format is:</p> <p>mmdyy</p> <p>or</p> <p>mmdyyyy</p> <p>-----</p> <p>no default value</p>	<p>for compression:</p> <p>pkzip25 -add -bef=091595 save.zip *.doc</p> <p>for extraction:</p> <p>pkzip25 -ext -bef=091595 save.zip *.doc</p>	<p>add, extract, delete, test, view</p>

Name/Description:	Value(s):	Example usage:	Used with:
<p><b>comment</b></p> <p>include a text comment for individual files within a .ZIP file</p> <p>configurable</p>	<p><b>all</b> - all files within .ZIP file</p> <p><b>unchanged</b> - only existing files that have not changed</p> <p><b>add</b> - only new files</p> <p><b>freshen</b> - only existing files</p> <p><b>update</b> - existing and new files</p> <p><b>none</b> - turn off comment</p> <p>-----</p> <p>default = <b>add</b></p>	<p>pkzip25 -add -com=all save.zip *.doc</p>	<p>add</p>
<p><b>configuration</b></p> <p>define default values for most PKZIP commands/options</p>	<p>any configurable command/option specified in the previous chapters</p> <p>-----</p> <p>no default value</p>	<p>pkzip25 -config -extract=freshen</p> <p>to see the current configuration values, type:</p> <p>pkzip25 -config</p>	<p>stand-alone</p>
<p><b>console</b></p> <p>perform a "mock" extract and display files on your screen</p>	<p>no suboptions</p> <p>-----</p> <p>no default value</p>	<p>pkzip25 -console save.zip *.txt</p>	<p>stand-alone</p>
<p><b>default</b></p> <p>reset the original defaults for the commands-options in the configuration file</p>	<p>no suboptions</p> <p>-----</p> <p>no default value</p>	<p>to reset all defaults:</p> <p>pkzip25 -default</p>	<p>stand-alone</p>

Name/Description:	Value(s):	Example usage:	Used with:
<p><b>delete</b></p> <p>remove files from a .ZIP file</p>	<p>file(s) to delete</p> <p>-----</p> <p>no default value</p>	<p>for individual files:</p> <pre>pkzip25 -del save.zip doc1.txt</pre> <p>for a specific file pattern:</p> <pre>pkzip25 -del save.zip *.doc</pre>	stand-alone
<p><b>directories</b></p> <p>store directory path names during compression, or recreate directory path names while extracting</p> <p>includes subdirectories (recurse)</p> <p>configurable</p> <p><b>note:</b> using this command is the same as combining the path and recurse commands</p>	<p><b>current</b> - store path information relative to the current path</p> <p><b>root or full</b> - store the entire path beginning at the root directory</p> <p><b>specify</b> - store the path information for each file being compressed (or recreates for each file being extracted), as specified on the command line</p> <p><b>relative</b> - store the directory path relative to the current working directory of the drive(s) specified (<b>DOS</b>)</p> <p><b>none</b> - overrides directory path information in configuration file</p> <p>-----</p> <p>default = <b>none</b></p> <p>default if used on command line without a sub-option = <b>current</b></p>	<p>compression example (assumes you are in "/wp"):</p> <pre>pkzip25 -add -dir=root save.zip wp/docs/*</pre> <p>the path stored would be "wp/docs/*"</p> <pre>pkzip25 -add -dir=current save.zip wp/docs/*</pre> <p>the path stored would be: "docs/*"</p> <p>extraction example:</p> <pre>pkzip25 -extract -directories save.zip /*</pre> <p><b>note:</b> UNIX users should utilize the <b>include</b> option or place quotation marks around wildcard designations to bypass automatic wildcard expansion by the shell, which may restrict your pattern search; see the <b>Getting Started Manual</b> for more information</p>	add, extract

Name/Description:	Value(s):	Example usage:	Used with:
<p><b>exclude</b></p> <p>exclude files from being compressed or extracted</p> <p>configurable</p> <p><b>note:</b> you must specify a sub-option (e.g. file pattern or list file name preceded by an appropriate list character "@" ) with the exclude option</p>	<p>the file(s) or file pattern (for example, *.doc) being excluded</p> <p>-----</p> <p>no default value</p>	<p>compression example:</p> <p>pkzip25 -add -excl=*.doc save.zip</p> <p>extraction example:</p> <p>pkzip25 -ext -excl=*.txt save.zip</p> <p>setting exclude default:</p> <p>pkzip25 -config -excl=*.txt</p> <p><b>note:</b> when you use the exclude option with the configuration command, PKZIP prompts you to configure the exclude default for add and/or extract operations</p>	<p>add, extract, delete, test, view</p>
<p><b>extract</b></p> <p>extract files from a .ZIP file</p> <p>configurable</p>	<p><b>all</b> - all files in .ZIP file</p> <p><b>freshen</b> - only files in the .ZIP file that exist in the target directory and that are "newer" than those files will be extracted</p> <p><b>update</b> - files in the .ZIP file which already exist in the target directory and that are "newer" than those files as well as files that are "not" in the target directory will be extracted</p> <p>-----</p> <p>default = all</p>	<p>pkzip25 -ext=up save.zip</p>	<p>stand-alone</p>
<p><b>fast</b></p> <p>set the level of compression to "fast" or level 2</p> <p>configurable</p>	<p>no suboptions</p> <p>-----</p> <p>no default value</p>	<p>pkzip25 -add -fast save.zip *.doc</p> <p>pkzip25 -config -fast</p>	<p>add</p>

Name/Description:	Value(s):	Example usage:	Used with:
<p><b>filetype</b></p> <p>processes files with the specified file type information in the .ZIP file</p> <p><b>(UNIX)</b></p> <p><b>note:</b> a "." before a filetype sub-option tells PKZIP to exclude the specified filetype(s) regardless of the default configuration setting (e.g. -filetype=hidden will exclude hidden files regardless of the default configuration setting)</p>	<p><b>hidden</b> - include/exclude filenames that have a dot "." in the first position of the filename (e.g. .profile)</p> <p><b>pipe</b> - include/exclude pipe files. These are files having a file mode starting with "p" (e.g. prwxrw-rw-)</p> <p><b>mlink</b> - include/exclude symbolic links. These are files having a file mode starting with "l" (e.g. lrw-rw-rw-)</p> <p><b>hlink</b> - include/exclude hard linked files. These are linked files that are not symbolic links. They are files with a link count &gt; 1</p> <p><b>none</b> - exclude all of the above file types; generally, this option should be followed by one, or more, file types. This results in just the type(s) specified being included in the ZIP file. For example, -filetype=none, pipe results in only PIPE files being included</p> <p><b>all</b> - include all of the above file types</p> <p><b>regular</b> - include/exclude regular files</p> <p>-----</p> <p>default = <b>regular</b></p>	<p>pkzip25 -add -filetype=all save.zip</p>	<p>add</p>

Name/Description:	Value(s):	Example usage:	Used with:
<p><b>fix</b></p> <p>recover a corrupt .ZIP file</p>	<p>no suboptions</p> <p>-----</p> <p>no default value</p>	<p>pkzip25 -fix save.zip</p>	<p>stand-alone</p>
<p><b>header</b></p> <p>create a comment for a .ZIP file, which will appear in the header area of the .ZIP file</p> <p>configurable</p>	<p>a text string or file specified with a list file character (e.g. @) that represents the header information</p> <p>-----</p> <p>no default value</p>	<p>to include specific text:</p> <p>pkzip25 -add -hea save.zip *.doc</p> <p><b>note:</b> when you type this command, PKZIP will prompt you for the header text</p> <p>to include an existing file:</p> <p>pkzip25 -add -hea=@text.doc save.zip *.doc</p>	<p>add, stand-alone</p>
<p><b>help</b></p> <p>display help screen for PKZIP</p>	<p>any command or option for which help is desired</p> <p>-----</p> <p>no default value</p>	<p>pkzip25 -help</p> <p>pkzip25 -help=add</p> <p>in this example you are specifying a specific command (add) for which you wish to view the help file</p>	<p>stand-alone</p>
<p><b>id</b></p> <p>preserve the user ID (UID) and/or group ID (GID) on extraction</p> <p><b>(UNIX)</b></p> <p><b>note:</b> the user who extracts files with preserved UID and GID information must have the same UID as is archived in the .ZIP file or root (superuser) file privileges</p>	<p><b>userid</b> - retain the user ID on extraction</p> <p><b>groupid</b> - retain the group ID on extraction</p> <p><b>all</b> - retain both the user ID and group ID on extraction</p> <p><b>none</b> - retain neither the user ID or group ID on extraction</p> <p>-----</p> <p>no default value</p>	<p>pkzip25 -extract -id=userid save.zip</p>	<p>extract</p>

Name/Description:	Value(s):	Example usage:	Used with:
<p><b>include</b></p> <p>include files to compress or extract</p> <p>configurable</p> <p><b>note:</b> you must specify a sub-option (e.g. file pattern or list file name preceded by an appropriate list character "@"*) with the exclude option</p>	<p>the file(s) or file pattern (for example, *.doc) being included</p> <p>-----</p> <p>no default value</p>	<p>pkzip25 -add -include=*.doc save.zip</p> <p>in this example, only .doc files will be compressed</p> <p>pkzip25 -config -include=*.txt</p> <p>in this example, you are setting up .txt files as the files that you always want to compress or extract, until you change the default or override from the command line with the exclude option</p> <p><b>note:</b> when you use the include option with the configuration command, PKZIP prompts you to configure the include default for add and/or extract operations</p>	<p>add, extract, delete, test, view</p>
<p><b>level</b></p> <p>set the level of compression</p> <p>configurable</p>	<p>any digit from 0 through 9, with 0 being no compression at the fastest speed, and 9 being the most compression at the slowest speed</p> <p>-----</p> <p>default = level <b>5</b> (normal)</p>	<p>pkzip25 -add -lev=9 save.zip *.doc</p>	<p>add</p>
<p><b>license</b></p> <p>display the product license information for PKZIP</p>	<p>no suboptions</p> <p>-----</p> <p>no default value</p>	<p>pkzip25 -lic</p>	<p>stand-alone</p>
<p><b>links</b></p> <p>specify that linked files be followed or preserved in a .ZIP archive</p> <p>(UNIX)</p> <p><b>note:</b> following a link results in a larger .ZIP file size since two copies of file data are compressed as though each link is a separate file</p>	<p><b>hlink</b> - hard links will be followed (stored) rather than preserved</p> <p><b>slink</b> - symbolic links will be followed (stored) rather than preserved</p> <p><b>all</b> - symbolic and hard links will be followed rather than preserved</p> <p><b>none</b> - symbolic and hard links will be preserved</p> <p>-----</p> <p>default = <b>none</b></p>	<p>pkzip25 -add -filetype=hlink -links=hlink save.zip</p> <p>this example compresses regular and hard linked files and duplicates link and file data for each hard linked file added to the .ZIP file</p>	<p>add, filetype</p> <p><b>note:</b> you must use both the add command and filetype option with the links option</p>

Name/Description:	Value(s):	Example usage:	Used with:
<p><b>listchar</b></p> <p>set the list character to the specified ASCII character for list files</p> <p>configurable</p>	<p>any valid single character</p> <p>-----</p> <p>default = @</p>	<p>pkzip25 -config -listchar=+</p>	<p>add, extract, delete, test, view</p>
<p><b>listsfxtypes</b></p> <p>display a list of the types of SFX files that can be created with PKZIP</p>	<p>no suboptions</p> <p>-----</p> <p>no default value</p>	<p>pkzip25 -listsfxtypes</p>	<p>stand-alone</p>
<p><b>locale</b></p> <p>change the system locale environment value that defines the valid formats for all date and time fields entered using PKZIP</p> <p>configurable</p> <p>(OS/2)</p>	<p>a valid country name (e.g. <b>us, germany, france</b>)</p> <p><b>environment</b> - PKZIP will attempt to use the environment variable defined by the operating system</p> <p>-----</p> <p>default = <b>US</b></p> <p>default if used on command line without a sub-option = <b>environment</b></p>	<p>pkzip25 -config -locale=germany</p> <p>pkzip25 -add -locale=germany test.zip *.doc</p> <p>pkzip25 -add -locale=environment test.zip *.doc</p>	<p>add, extract</p>
<p><b>locale</b></p> <p>set the default PKZIP time and date settings to match your system time and date formats</p> <p>configurable</p> <p>(UNIX, DOS)</p>	<p>no suboptions</p> <p>-----</p> <p>default = (time=12-Hour; date=MMDDYY)</p>	<p>pkzip25 -config -locale</p> <p>pkzip25 -add -locale test.zip *.doc</p>	<p>add, extract</p>
<p><b>lowercase</b></p> <p>extracts file name(s) in lower case regardless of how it was originally archived</p> <p>(UNIX)</p>	<p>no suboptions</p> <p>-----</p> <p>no default value</p>	<p>pkzip25 -extract -lowercase save.zip *</p>	<p>extract</p>

Name/Description:	Value(s):	Example usage:	Used with:
<p><b>mask</b></p> <p>remove file attributes of files within a .ZIP file or when extracting</p> <p><b>note:</b> you can only remove the attributes that have been stored, as defined by the attribute command</p> <p>configurable</p> <p><b>(DOS, OS/2)</b></p>	<p><b>hidden</b> - hidden attributes</p> <p><b>system</b> - system attributes</p> <p><b>readonly</b> - read-only attributes</p> <p><b>none</b> - no attributes (turns off attribute mask in the PKZIP Configurations Settings file for this instance only)</p> <p><b>all</b> - all attributes</p> <p>-----</p> <p>default (compress) = <b>none</b></p> <p>default (extract) = <b>all</b></p> <p>default if used on command line without a sub-option (compress and extract) = <b>all</b></p>	<p>pkzip25 -add -attr=all -mask=hidden save.zip</p> <p>pkzip25 -extract -mask=none save.zip</p> <p>pkzip25 -config -mask=hidden</p> <p><b>note:</b> when you use the mask option with the configuration command, PKZIP prompts you to configure the mask default for add and/or extract operations</p>	<p>add, extract</p>
<p><b>maximum</b></p> <p>set the level of compression to the highest level, but lowest speed</p> <p>configurable</p>	<p>no suboptions</p> <p>-----</p> <p>no default value</p>	<p>pkzip25 -add -max save.zip *.doc</p> <p>pkzip25 -config -max</p>	<p>add</p>

Name/Description:	Value(s):	Example usage:	Used with:
<p><b>more</b></p> <p>pause after one screen of output and prompt to continue</p> <p>configurable</p>	<p>the number of rows of information you want to consist of a screen</p> <p>-----</p> <p>default = one screen of information</p>	<p>pkzip25 -view -more=22 save.zip</p> <p>pkzip25 -config -more</p> <p><b>note:</b> when you use the more option with the configuration command, PKZIP prompts you to configure the more default for add, extract, and/or view operations</p>	all commands
<p><b>move</b></p> <p>remove the files from the source drive after compression (files will reside only in .ZIP file)</p> <p>configurable</p>	<p>none</p> <p>-----</p> <p>no default value</p>	<p>pkzip25 -add -move save.zip *.doc</p> <p>pkzip25 -config -move</p> <p><b>note:</b> if the move option is enabled by default in the configuration file, PKZIP will display a confirmation prompt every time you attempt to modify the configuration file</p>	add
<p><b>namesfx</b></p> <p>specify a file name when converting to a self-extracting file</p>	<p>a valid file name (e.g. filename.exe)</p> <p>-----</p> <p>no default value</p>	<p>pkzip25 -sfx -namesfx=test.exe docs.zip</p>	sfx
<p><b>nametype</b></p> <p>specify the format in which you wish to extract file(s)</p> <p>configurable</p> <p>(OS/2)</p>	<p><b>auto</b> - auto-detection of file system and extraction format</p> <p><b>short</b> - files are extracted in 8+3 format</p> <p><b>long</b> - files are extracted in same format that they were originally added in</p> <p>-----</p> <p>default = <b>auto</b></p> <p>default if used on command line without a sub-option = <b>short</b></p>	<p>pkzip25 -extract -nametype=long test.zip /temp</p>	extract
<p><b>noextended</b></p> <p>suppress the storage (or extraction) of extended attribute information within the .ZIP file</p> <p>configurable</p>	<p>no suboptions</p> <p>-----</p> <p>no default value</p>	<p>pkzip25 -add -noextended save.zip *</p>	<p>add, extract</p> <p><b>note:</b> the extract command works with the noextended option on (DOS) only</p>

Name/Description:	Value(s):	Example usage:	Used with:
<p><b>nofix</b></p> <p>suppress the "attempt to fix" prompt if PKZIP encounters errors in a .ZIP file</p> <p>configurable</p>	<p>no suboptions</p> <p>-----</p> <p>no default value</p>	<p>pkzip25 -add -nofix save.zip *.doc</p>	<p>add, extract, delete, test, view</p>
<p><b>normal</b></p> <p>set the level of compression to 5, the best balance of compression and speed</p> <p>configurable</p>	<p>no suboptions</p> <p>-----</p> <p>no default value</p>	<p>pkzip25 -add -normal save.zip</p> <p>pkzip25 -config -normal</p>	<p>add</p>
<p><b>nozipextension</b></p> <p>suppress the adding of a .ZIP extension to the .ZIP file name</p> <p>configurable</p>	<p>no suboptions</p> <p>-----</p> <p>no default value</p>	<p>pkzip25 -add -nozipextension file.ibm *.doc</p>	<p>all commands</p>
<p><b>optionchar</b></p> <p>set the option character to the specified ASCII character</p> <p>configurable</p>	<p>any valid single character</p> <p>-----</p> <p>default = -</p>	<p>pkzip25 -opt=+ +add save.zip *.doc</p> <p>pkzip25 -config -option=+</p>	<p>all commands</p>

Name/Description:	Value(s):	Example usage:	Used with:
<p><b>overwrite</b></p>	<p><b>prompt</b> - prompt every file individually on whether to overwrite a file that has the same name as the one being extracted</p> <p><b>all</b> - overwrite all files that have a corresponding file on the hard drive</p> <p><b>never</b> - never overwrite a file that has a corresponding file on the hard drive</p> <p>-----</p> <p>default = <b>prompt</b></p> <p>default if used on command line without a sub-option = <b>all</b></p>	<p>pkzip25 -ext -over=all save.zip</p>	<p>extract</p>
<p><b>password</b></p> <p>include a password with your .ZIP file</p> <p>configurable</p>	<p>your password, or no value</p> <p>-----</p> <p>no default value</p>	<p>to include password in the command:</p> <p>pkzip25 -add -pass=beowulf save.zip</p> <p>to have PKZIP prompt you for a password "after" you type the command:</p> <p>pkzip25 -add -pass save.zip</p>	<p>add, extract, test</p>

Name/Description:	Value(s):	Example usage:	Used with:
<p><b>path</b></p> <p>store directory path names for files within a .ZIP file</p> <p>configurable</p>	<p><b>current</b> - stores the path from the current directory</p> <p><b>root/full</b> - store the entire path beginning at the root of the drive; also referred to as "full" path</p> <p><b>specify</b> - store the path as specified in your pkzip command</p> <p><b>relative</b> - store the directory path relative to the current working directory of the drive(s) specified (<b>DOS</b>)</p> <p><b>none</b> - no path information stored</p> <p>-----</p> <p>default = <b>none</b></p> <p>default if used on command line without a sub-option = <b>current</b></p>	<p>assuming in you are in "/temp":</p> <p>pkzip25 -add -path=root save.zip docs/*</p> <p>(the complete path is stored including "temp/docs/")</p> <p>pkzip25 -add -path=current save.zip docs/wp*</p> <p>(the path stored would be "docs/wp")</p>	add
<p><b>permission</b></p> <p>preserve the original file access permissions of an extracted file</p> <p>(UNIX)</p> <p><b>note:</b> the user who extracts files with preserved UID and GID information must have the same UID as is archived in the .ZIP file or root (superuser) file privileges</p>	<p>no suboptions</p> <p>-----</p> <p>no default value</p>	pkzip25 -extract -permission save.zip	extract
<p><b>preview</b></p> <p>preview the results of a set of commands or options without actually performing the task represented by that command/option</p>	<p>no suboptions</p> <p>-----</p> <p>no default value</p>	pkzip25 -add -preview save.zip	add

Name/Description:	Value(s):	Example usage:	Used with:
<p><b>print</b></p> <p>print a file within a .ZIP file</p> <p>(DOS, OS/2)</p>	<p>the print device use, for example, "lpt1"</p> <p>-----</p> <p>default = the default printer on your system</p>	<p>pkzip25 -print=lpt1 save.zip readme.txt</p> <p>if you do not specify a print device, the "default" printer is used</p>	stand-alone
<p><b>recurse</b></p> <p>search subdirectories for files to compress</p> <p>configurable</p>	<p>no suboptions</p> <p>-----</p> <p>no default value</p>	<p>pkzip25 -add -rec save.zip *</p> <p><b>note:</b> UNIX users should utilize the <b>include</b> option or place quotation marks around wildcard designations to bypass automatic wildcard expansion by the shell, which may restrict your pattern search; see the <b>Getting Started Manual</b> for more information</p>	add
<p><b>sfx</b></p> <p>create a self-extracting .ZIP file (.exe file), or convert an existing .ZIP file to a .exe file</p> <p>configurable</p> <p><b>note:</b> for a listing of available self extractors, use the listsfxtypes option</p>	<p><b>dosfull</b> - create a DOS-formatted self-extractor</p> <p><b>jrds</b> - create a DOS junior self-extractor</p> <p><b>win16</b> - create a Windows 16-bit self-extractor</p> <p>-----</p> <p>default = (create an sfx native to the operating system)</p>	<p>for a new .ZIP file:</p> <p>pkzip25 -add -sfx save</p> <p>to convert an existing .ZIP file:</p> <p>pkzip25 -sfx save.zip</p>	add, stand-alone
<p><b>shortname</b></p> <p>convert .ZIP file name in long file name format to a DOS or OS/2 equivalent "short" file name</p> <p><b>note:</b> PKZIP includes sub-options for both DOS and OS/2 because of the differences in the way each OS handles short file names</p> <p>configurable</p>	<p><b>dos</b> - convert .ZIP file to a dos-equivalent short file name</p> <p><b>os2</b> - convert .ZIP file to an OS/2 equivalent short file name</p> <p>-----</p> <p>no default value</p>	<p>pkzip25 -add -short=dos save.zip</p>	add

Name/Description:	Value(s):	Example usage:	Used with:
<p><b>silent</b></p> <p>suppress all screen display output</p> <p>configurable</p> <p><b>note:</b> errors and warnings are displayed whether the silent option is specified or not, however, prompts for other PKZIP operations are not displayed (e.g. Password)</p>	<p>no suboptions</p> <p>-----</p> <p>no default value</p>	<p>pkzip25 -add -silent save.zip *.doc</p> <p>pkzip25 -config -silent</p> <p><b>note:</b> if the silent option is enabled by default in the configuration file, PKZIP will display a confirmation prompt every time you attempt to modify the configuration file</p>	<p>all commands</p>
<p><b>sort</b></p> <p>sort files within .ZIP file based on specific criteria (for example, by file size)</p> <p>configurable</p> <p><b>note:</b> the crc and ratio suboptions will not work with the add command and sort option</p>	<p><b>crc</b> - sort by CRC value</p> <p><b>date</b> - sort by file date of file</p> <p><b>extension</b> - sort by file extension</p> <p><b>name</b> - sort alphabetically by name</p> <p><b>natural</b> - sort in the order that the file was compressed</p> <p><b>ratio</b> - sort by compression ratio</p> <p><b>size</b> - sort by the original, uncompressed size of the file ("length" in display)</p> <p><b>none</b> - do not sort (same as natural)</p> <p>-----</p> <p>default = <b>natural</b></p> <p>default if used on command line without a sub-option = <b>name</b></p>	<p>pkzip25 -add -sort=date save.zip *.doc</p> <p>pkzip25 -config -sort=date</p> <p><b>note:</b> when you use the sort option with the configuration command, PKZIP prompts you to configure the mask default for add, extract, and/or view operations</p>	<p>add, extract, test, view</p>

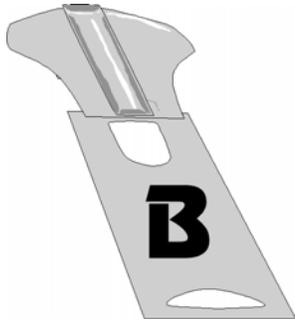
Name/Description:	Value(s):	Example usage:	Used with:
<p><b>span</b></p> <p>forces disk spanning in the event that PKZIP is unable to detect the fact that destination directory is on removable media</p> <p><b>note:</b> the spanning process should proceed automatically; a command line option is usually not necessary to enable this feature</p> <p>(DOS, OS/2)</p>	<p>no suboptions</p> <p>-----</p> <p>no default value</p>	<p>pkzip25 -add -span a:\save.zip *.doc</p>	<p>add</p>
<p><b>speed</b></p> <p>sets the level of compression to 1, which is the least but fastest compression</p> <p>configurable</p>	<p>no suboptions</p> <p>-----</p> <p>no default value</p>	<p>pkzip25 -add -speed save.zip *.doc</p> <p>pkzip25 -config -speed</p>	<p>add</p>
<p><b>store</b></p> <p>sets the level of compression to 0 (no compression), and only stores the files within the .ZIP file</p> <p>configurable</p>	<p>no suboptions</p> <p>-----</p> <p>no default value</p>	<p>pkzip25 -add -store save.zip *.doc</p> <p>pkzip25 -config -store</p>	<p>add</p>
<p><b>temp</b></p> <p>creates the temporary .ZIP file required by PKZIP, on specified drive</p> <p><b>note:</b> the temp option only works when updating an existing .ZIP file</p>	<p>the drive and/or path.</p> <p>e.g. C: or /root/temp</p> <p>-----</p> <p>no default value</p>	<p>pkzip25 -add -temp=z:\public test.zip *.txt</p> <p>this example updates the .ZIP file test.zip and uses the z:\public directory location for the temp file</p> <p>pkzip25 -add -temp=/temp test.zip *.txt</p> <p>this example updates the .ZIP file test.zip and uses the /temp directory location for the temp file</p>	<p>add</p>

Name/Description:	Value(s):	Example usage:	Used with:
<p><b>test</b></p> <p>test the integrity of files within a .ZIP file</p> <p>configurable</p>	<p><b>all</b> - all files in .ZIP file will be tested</p> <p><b>freshen</b> - only those files in the .ZIP file which already exist in the extract directory and that have a file modification date newer than the files already in the directory will be tested</p> <p><b>update</b> - files in the .ZIP file which already exist in the target directory and that are "newer" than those files and files that are "not" in the target directory will be tested</p> <p>-----</p> <p>default = <b>all</b></p>	<p>pkzip25 -test save.zip</p>	<p>stand-alone</p>
<p><b>times</b></p> <p>specify whether file access, creation, and modification times are to be preserved</p> <p><b>(DOS, UNIX)</b></p>	<p><b>access</b> - restores the time of last access to file(s) on extraction</p> <p><b>modify</b> - restores the time of last modification to files on extraction</p> <p><b>create</b> - restores the time of creation to files on extraction <b>(DOS)</b></p> <p><b>all</b> - all file times are restored</p> <p><b>none</b> - file times are not restored</p> <p>-----</p> <p>default = <b>none</b></p>	<p>pkzip25 -extract -times=access save.zip</p>	<p>extract</p>

Name/Description:	Value(s):	Example usage:	Used with:
<p><b>version</b></p> <p>determine the version of PKZIP you are running</p> <p><b>note:</b> PKZIP also returns the version number to the shell by way of setting the error level; the version number value is passed directly to the shell and as such is not viewable by the user</p>	<p><b>major</b> - returns the major version number of the release</p> <p><b>minor</b> - returns the minor number of the release (e.g. - if the version number is 2.5.1, the value returned is 5)</p> <p><b>step</b> - returns the step, or patch value. (e.g. - if the version is 2.04.01, the step value returned is 1)</p> <p>-----</p> <p>default = <b>major</b></p>	<p>pkzip25 -version</p>	<p>stand-alone</p>
<p><b>view</b></p> <p>display information about the files within a .ZIP file, for example, the compressed size of a file</p> <p>configurable</p>	<p><b>brief</b> - present information in the most compact manner</p> <p><b>details</b> - present information in the most detailed manner</p> <p><b>normal</b> - present information in the normal manner</p> <p>-----</p> <p>default = <b>normal</b></p>	<p>pkzip25 -view save.zip</p>	<p>stand-alone</p>
<p><b>volume</b></p> <p>store/restore the volume label with the .ZIP file</p> <p>configurable</p> <p><b>(DOS, OS/2)</b></p>	<p>a drive letter (for example, C)</p> <p>-----</p> <p>default = no volume label</p> <p>default if used on command line without a sub-option = current drive</p>	<p>pkzip25 -add -volume=C save.zip *.docs</p>	<p>add, extract</p>
<p><b>warning</b></p> <p>pause after every warning, and prompt you to continue or not</p> <p>configurable</p>	<p>no suboptions</p> <p>-----</p> <p>no default value</p>	<p>pkzip25 -ext -warn save.zip *</p>	<p>add, extract, test, view</p>

Name/Description:	Value(s):	Example usage:	Used with:
<p><b>zipdate</b></p> <p>set the file modification date of the .ZIP file</p> <p>configurable</p>	<p><b>newest</b> - set to the date of the "newest" file within the .ZIP file</p> <p><b>oldest</b> - set to the date of the "oldest" file within the .ZIP file</p> <p><b>retain</b> - retain the original date of the .ZIP file (the date on which the .ZIP file was created)</p> <p><b>none</b> - disable the .ZIP file date in the configuration file, and set the .ZIP date as the last modification date</p> <p>-----</p> <p>default = the current date</p> <p>default if used on command line without a sub-option = <b>retain</b></p>	<p>pkzip25 -add=update -zipdate=retain save.zip *.txt</p>	<p>add</p>





## Appendix B Error and Warning Messages

This appendix contains reference information on all error and warning messages that can occur in PKZIP. An error usually causes the canceling of the task you are performing, for example, compressing a file. A warning usually indicates that something is wrong, but it is not severe enough to cancel an entire task. It might also be a reminder or query prompt.

### Error Messages

---

When an error occurs, PKZIP displays an error message. The following is a description of each error message.

Error:	Potential Cause(s):
<b>(E2)</b> Ambiguous option or command specified - XXXX	If you abbreviate an option on your command line, make sure that you are supplying enough characters in the option to delineate it from similarly spelled options. If, for example, you only specify <b>-pr</b> on your command line, PKZIP will generate the (E2) error because it cannot determine whether you are specifying the <b>print</b> or <b>preview</b> option.

Error:	Potential Cause(s):
<b>(E3)</b> Ambiguous sub-option specified - XXXX	If you abbreviate a sub-option on your command line, make sure that you are supplying enough characters in the sub-option to delineate it from similarly spelled sub-options. If, for example, you only specify <code>-sort=na</code> on your command line, PKZIP will generate the (E3) error because it cannot determine whether you are specifying the name or natural sub-option.
<b>(E4)</b> Unknown or illegal option - XXXX	The option you specified on the command line is invalid. It does not match any known options. Verify that you typed the option correctly. Check the spelling.
<b>(E5)</b> Unknown or illegal sub-option - XXXX	The sub-option you specified on the command line is invalid. It does not match any known sub-options. Verify that you typed the sub-option on your command line correctly. Verify that you are not using an illegal sub-option (e.g. <code>-add -sort=crc</code> ). Check the spelling.
<b>(E6)</b> No .ZIP file specified	There was not a .ZIP file specified on the command line. PKZIP does not accept wildcards for .ZIP file name when adding files to a .ZIP archive.
<b>(E7)</b> Can't create: XXXX	PKZIP couldn't create a .ZIP file when fixing. PKZIP couldn't create a volume label on a spanned archive. PKZIP couldn't create a temporary file for a spanned archive. Verify that you have write access to the drive or diskette on which you are creating these files.
<b>(E8)</b> Nothing to do!	You didn't do something that is required for a particular task. For example, PKZIP couldn't find the file you are trying to open or access (you might have specified to update a pattern such as <code>*.txt</code> and PKZIP didn't find any files that matched or that needed updating).

Error:	Potential Cause(s):
<b>(E9)</b> No file(s) found	PKZIP can't find the file you are trying to access. For example, you might be trying to extract files from a .ZIP archive that don't exist in that archive. Verify that the file(s) you specify on the command line exactly match the file(s) in the .ZIP file. If, for example, the file in the .ZIP archive is stored with path information and you attempt to extract it, but specify the file name only, you will get the (E9) error.
<b>(E10)</b> No files specified for deletion	There are no files or file patterns specified for deletion on the command line. In lieu of a specified file or file pattern, PKZIP will not assume that the user wishes to specify all (*) files.
<b>(E11)</b> Disk full, file: XXXX	The hard disk or floppy disk you are writing to is full. This error occurs when PKZIP attempts to write a .ZIP file, or extract a file contained in a .ZIP file to a hard or floppy disk that is full. Free up sufficient disk space and try again.
<b>(E12)</b> Can't find .ZIP file: XXXX	PKZIP can't find the .ZIP file you specified. This error will only occur when you use commands/options/sub-options that work with existing .ZIP files. Verify that the file is specified correctly. If you are adding files to an archive, verify that you place the .ZIP file name before specifying files to be added on the command line. If the .ZIP file is not in the same directory where you typed the command, make sure to include path information.  (e.g. pkzip25 -add=freshen /temp/test.zip *.txt)
<b>(E13)</b> Can't open .ZIP file: XXXX	The named .ZIP file is read-only or locked by another application and can not be modified. This may also occur on a Network drive if you do not have sufficient access rights to the file to allow you to modify it.

Error:	Potential Cause(s):
<b>(E14)</b> Can't create .ZIP file: XXXX	PKZIP is not able to create the .ZIP file. Verify that the destination directory is not full, the .ZIP file does not already exist, or is not read-only. If you are creating the file on a network drive, confirm that you have the appropriate rights to the network file system.
<b>(E15)</b> Renaming temporary .ZIP file, saved as: XXXX	PKZIP could not rename the temporary file to the specified .ZIP file name. Verify that the destination drive is not full. If you are updating a non-spanned .ZIP file on removable media (e.g. floppy diskette) and the updated archive exceeds the size available on the removable media, you will receive the (E15) error. You will need to recreate the archive for spanning. Keep in mind that you cannot update a spanned archive. If you are creating the file on a network drive, confirm that you have the appropriate rights to the network file system.
<b>(E16)</b> Can't open for write access, file: XXXX	PKZIP is unable to write to the specified file or device. Verify that you have write access to the file or that your printer is configured correctly. Additionally if you are using the PKSFXS.DAT file, verify that you have PKSFADATA environment variable configured correctly. See page 80 for more information.
<b>(E17)</b> Error encrypting file data	PKZIP encountered a problem with the compressed data that it was trying to encrypt. For example, the disk on which the compressed data was located was bad or corrupt.
<b>(E18)</b> Can't open list file: XXXX	The named list file could not be found. It does not exist, was spelled incorrectly, isn't located in the specified directory, or can't be accessed because the user does not have the appropriate rights to the file.
<b>(E19)</b> Aborted file extract	Extraction process was terminated by the user while changing disks during a disk spanning operation.
<b>(E20)</b> Aborted file compression	Compression process was terminated by the user while changing disks during a disk spanning operation.

Error:	Potential Cause(s):
<b>(E21)</b> Can't modify a spanned .ZIP file	Spanned .ZIP files cannot be modified. The spanned archive will need to be recreated.
<p><b>(E100)</b> Insufficient memory - To create input work buffer</p> <p><b>(E101)</b> Insufficient memory - To create .ZIP work buffer</p> <p><b>(E102)</b> Insufficient memory - To create .ZIP file: XXXX</p> <p><b>(E103)</b> Insufficient memory - .ZIP central header data</p> <p><b>(E104)</b> Insufficient memory - For input file list</p> <p><b>(E105)</b> Insufficient memory - To extract files from .ZIP</p> <p><b>(E106)</b> Insufficient memory - To fix .ZIP file</p> <p><b>(E107)</b> Insufficient memory - To create SFX file</p> <p><b>(E108)</b> Insufficient memory - For Authenticity Verification</p>	<p>Insufficient memory is available to process the .ZIP file. Try making more memory available to PKZIP. If this does not rectify the problem then the .ZIP file may be corrupted. The -fix option may correct the problem. If you create a new .ZIP file and receive this message, it may be due to the number of files you are attempting to compress. Reduce the scope of your command and try again. If you are using a LIST file in your PKZIP command, you may be receiving this error because the LIST file is too large. See page 54 for more information on LIST files.</p>
<b>(Z150)</b> Error reading .ZIP file	PKZIP can't read the .ZIP file or is unable to read the central directory record. The file might be located on a corrupt disk or part of a disk. This includes floppy disks.
<b>(Z151)</b> File too small for valid .ZIP data	The .ZIP file is too small to be considered a valid .ZIP file. The .ZIP header must be a minimum number of bytes. The .ZIP file PKZIP encountered was smaller than the minimum.
<b>(Z152)</b> No CE signature found	PKZIP is unable to locate the central end record in the .ZIP file. The file is not a valid .ZIP archive or has been corrupted. The fix option may repair the .ZIP file.

Error:	Potential Cause(s):
<b>(Z153)</b> Premature end of .ZIP	PKZIP reached the end of compressed data before anticipated. The archive may be damaged. The fix option may repair the .ZIP file.
<b>(Z154)</b> Local header corrupt or not found	PKZIP detected an error in the local header of the .ZIP file. The local header record could not be found or perhaps has an invalid date. The archive may be damaged. The fix option may repair the .ZIP file.
<b>(Z155)</b> Too many files in XXXX	PKZIP cannot add or extract files in excess of the limit of 65,535 files. (16,383 with the <b>204</b> option enabled) Reduce the number of files you are trying to process.
<b>(E255)</b> User pressed ctrl-c or control-break	This error occurs when you press CTRL+BREAK or CTRL+C in the middle of a PKZIP operation.

## Warning Messages

---

Sometimes a condition occurs that might cause a task to pause temporarily. This could be something that prevents part of a task from happening, or simply a message or reminder. For several of these conditions, PKZIP displays a warning message.

The following is a description of each warning message:

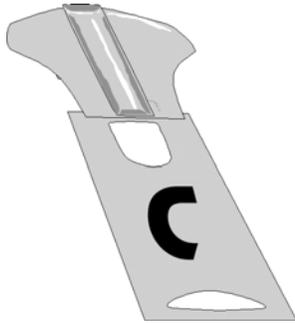
PKZIP Warning:	Potential Cause(s):
<b>(W1)</b> Can't create: XXXX	PKZIP could not create volume label, file, or directory. Verify that you have appropriate access rights to the file or directory.
<b>(W2)</b> Illegal path or drive specified: XXXX	The file being extracted has an invalid name or path. Verify that you have entered the correct path in your command line and that the file does not contain any inappropriate characters such as a colon or leading slash.

PKZIP Warning:	Potential Cause(s):
<b>(W3)</b> requires PKZIP version...	The .ZIP file is corrupt or the specified file is compressed or encoded in a way that this version of PKZIP is unable to handle. A different version of PKZIP may be required to properly extract this file. Contact PKWARE for more information.
<b>(W4)</b> File fails CRC check	It is likely that the file PKZIP is trying to extract is corrupt, and was not extracted correctly. For more information, see the CRC section in Appendix D - How Does PKZIP Work? on page 169.
<b>(W5)</b> Unexpected size in .ZIP file	PKZIP determined that the stored size of the .ZIP file does not equal the final size of the file being extracted.
<b>(W6)</b> Insufficient memory for .ZIP header comment - Header not set	This error occurs when PKZIP attempts to allocate memory for a new header and cannot. Try making more memory available to PKZIP.
<b>(W7)</b> file: XXXX already exists. Overwrite (<Y>es/<N>o/<A>ll/ne<V>er/<R>ename/<Esc>)?	The file(s) you are trying to extract already exists in the location to which you are extracting. By default, PKZIP prompts you before overwriting a file.
<b>(W8)</b> Could not open file: XXXX	You may not have the proper permissions to access the file or the file may have been locked by another program while PKZIP was trying to access it. If the file is located on a network file system, consult your System Administrator to verify your access rights.
<b>(W9)</b> Could not delete file: XXXX	You do not have the proper permissions to access and delete the file, or another application has the file open. This warning only occurs when the move option is used on the command line.
<b>(W10)</b> Deflated file has bad table <b>(W11)</b> Deflated file has bad compression block type	The deflated file has a bad table or compression block type. The file is probably corrupt or not a .ZIP file. Files that have been damaged in this way cannot be recovered.
<b>(W12)</b> Unexpected end of compressed data	Corrupt data caused PKZIP to abort the extraction before it could finish.

PKZIP Warning:	Potential Cause(s):
<b>(W13)</b> Skipping encrypted file: XXXX	PKZIP encountered a file that has been password protected. You need the password to access this file.
<b>(W14)</b> Imploded file has bad table <b>(W15)</b> Imploded file has bad literal table	The Imploded file being tested or extracted has an error in its encoding. The file is probably corrupt or not a .ZIP file. Files that have been damaged in this way cannot be recovered.
<b>(W16)</b> Can't extract reduced files	The compression method used to originally compress the file(s) (reduce) is not supported by PKZIP, therefore, the file(s) cannot be extracted.
<b>(W17)</b> Can't extract tokenized files	The compression method used to originally compress the file(s) (tokenize) is not supported by PKZIP, therefore, the file(s) cannot be extracted.
<b>(W18)</b> Unknown compression method for file: XXXX	An unfamiliar compression method has been used with the current .ZIP file.
<b>(W19)</b> Could not clear archive attribute on file: XXXX	PKZIP could not clear the archive attribute on a file. The file will be compressed but the archive bit cannot be cleared. This warning usually occurs when the add=incremental option is used on the command line.
<b>(W20)</b> Incorrect password for file: XXXX	Verify that you entered the correct password for the .ZIP file. When a .ZIP file is password protected, you can only access the contents of that .ZIP file with an appropriate password.  <b>Note:</b> Passwords are case sensitive. See the Including a Password section on page 69 for more information.
<b>(W21)</b> Temporary file created in current directory	PKZIP creates a temporary file for the file(s) being compressed when updating a .ZIP file. PKZIP is unable to create the temporary .ZIP file in the specified location. The file is placed into the current directory.

PKZIP Warning:	Potential Cause(s):
<b>(W22)</b> Authenticity Verification Failed!	The Authenticity Verification (AV) information contained in the .ZIP file is corrupt. Failure of AV indicates a file that has been tampered with or damaged. If the file has failed the AV check, the contents are suspect.
<b>(W23)</b> Authenticity Verification Failed!	The stored Authenticity Verification (AV) checksum value did not match the calculated checksum value. The .ZIP file has been tampered with or is perhaps corrupt.
<b>(W24)</b> file: XXXX short name already exists. Increment (<Y>es/<N>o/<A>ll/ne<V>er/<Esc>)?	<p>Before proceeding with file compression and .ZIP archive creation in an add operation, PKZIP creates a listing of all of the files to be added to the archive. This file list is created in the background and is not apparent to the user. PKZIP eventually uses this file list to create the .ZIP archive. When the shortname option is used on the command line, PKZIP attempts to shorten any long file names to the 8+3 format before adding it to this file list. The criteria PKZIP uses to shorten the file name is dependent on the specified operating system sub-option (e.g. os2, dos). There may be instances where PKZIP attempts to shorten two different long file names to the same short file name. If, for example, a shortened file name is added to the file list as:</p> <p>filename.txt</p> <p>If PKZIP subsequently attempts to rename a second file to a specific short name before adding it to the file list, but detects that the shortname already exists in the file list, it will generate the (W24) warning prompting you to increment the file name's truncated number. Assuming the shortname sub-option was set to os2, entering Y at the warning prompt would rename the file to something similar to the following:</p> <p>filename.txt</p> <p>After all of the specified files have been added to the file list, PKZIP will proceed with file compression and create the .ZIP archive.</p>

PKZIP Warning:	Potential Cause(s):
<p><b>(W25)</b> Distribution license information was not installed</p>	<p>The distribution license information was not correctly placed at the time of installation. The distribution license information is placed in a file called pkzip.ex_. If you need to re-install PKZIP, first verify that pkzip.ex_ does not exist in the installation directory. If pkzip.ex_ does exist in the installation directory, either delete the file or allow the setup program to overwrite the file. Do not rename pkzip.ex_ if prompted by the setup program.</p>
<p><b>(W26)</b> directory: XXXX already exists. Overwrite (&lt;Y&gt;es/&lt;N&gt;o/&lt;A&gt;ll/ne&lt;V&gt;er/&lt;Esc&gt;)?</p>	<p>Assuming the overwrite option is set to prompt, this warning appears when PKZIP attempts to extract a directory over an existing directory with the same name. Answering Y at this prompt will update any extended attributes (EAs) stored in the .ZIP file.</p>
<p><b>(W27)</b> Insufficient memory for extended attributes</p>	<p>This error occurs when PKZIP attempts to allocate memory for extended attributes and cannot. Try making more memory available to PKZIP.</p>



## Appendix C Frequently Asked Questions

This section lists some commonly asked questions about PKZIP and related subjects. We hope you will find this information helpful.

***Why do I get the message "SYS1041: The name specified is not recognized as an internal or external command, operable program or batch file." or "Bad command or file name" or "XXXX: not found"?***

---

These messages tell you that your operating system cannot find the program to which you are referring. This occurs because you are either not spelling the name of the program correctly, or you didn't put a space between the program name and its options, or the program has not been properly installed. If you are trying to run PKZIP and you get this error, it may be because pkzip25.exe is not in your search path.

***Why didn't the files I zipped get any smaller?***

---

On occasion, you may find that the files you add to a .ZIP file do not compress. These files are "stored". This occurs when a file is either already compressed or encrypted. You will often find that files distributed with commercial applications are already compressed.

### ***I zipped up a bunch of files but now I have LESS disk space?***

---

When PKZIP compresses files, it makes a copy of the original file. The original file(s) still exist. If you wish to recover space that was taken up by the original file(s), you must either delete them yourself, or instruct PKZIP to delete the file(s) with the **move** option.

### ***What's the difference between add=freshen and add=update?***

---

The **freshen** and **update** sub-options are very similar. This may be confusing at first, but the difference between them is easy to understand.

The **freshen** option tells PKZIP to archive any files which match those already in the .ZIP file. These files are re-compressed only if they are newer than the files already in the .ZIP file. Each file is evaluated individually.

The **update** sub-option archives all files, with one distinction. If the **update** option is not used, all files specified are compressed and added to the .ZIP file, even if they already exist in the .ZIP file. By using the **update** sub-option, you instruct PKZIP to compare what is already in the .ZIP file against what it was asked to compress. If a file is already present in the .ZIP file as well as the source directory, PKZIP compresses a file only if it is newer than the copy of the file within the .ZIP file. If a file in the source directory is not already present in the target .ZIP file, PKZIP adds it to the .ZIP file.

### ***Is PKZIP compression "lossy" or "lossless"?***

---

PKZIP uses a "lossless" compression scheme. This means that 100% of the original data is preserved and re-created. There is no difference between the data that you put in and the data that you get back out.

There are other compression methods that are known as "lossy". The idea behind these compression methods is that if you throw away some of the data, it becomes less complex and therefore can be compressed more. This type of compression is only useful for data that need not be precise. This applies to some applications that use pictures and sound.

### ***How do I include subdirectory information in my .ZIP file?***

---

In order to include subdirectory information in your .ZIP file, you must recurse the subdirectories and preserve path names. This is done with the **directories** option. For example:

```
pkzip25 -add -directories test.zip *
```

In this example, the current directory as well as all subdirectories and files contained therein are archived in a file called test.zip.

When a .ZIP file is created with paths stored, these paths are visible in a view of the file (**view**).

To re-create these subdirectories, or to place files into their original subdirectories, the **directories** option must be used with the **extract** command.

### ***I zipped up some subdirectories, but I can't get them to come back?***

---

Did you remember to use the **directories** option when you originally created the .ZIP file? Did you use the **directories** option when you extracted the contents of your .ZIP file? To verify that there are paths in the .ZIP file, do a view of the file:

```
pkzip25 -view test.zip
```

If you do not see paths as part of the file names within the .ZIP file, then paths are not stored and therefore cannot be recovered. If you do see paths make sure that you are using the **directories** option when you extract the files. For example:

```
pkzip25 -extract -directories test.zip
```

### ***How do I unzip a single file that is in a subdirectory in the .ZIP file?***

---

Type **pkzip25 -extract** with the name of the .ZIP file and the name of the particular file you want. With a .ZIP file that contains paths, the procedure is the same.

Let's assume you are working a file called test.zip the contains the following files:

```
file1.txt
temp/file2.txt
temp/tut/file3.txt
```

To extract only "file3.txt" from this .ZIP file, you must specify the complete name and path.

```
pkzip25 -extract test.zip temp/tut/file3.txt
```

If you wanted to extract it with its subdirectory, simply include the **directories** option on the command line.

### ***How do I unzip a directory without also extracting its subdirectories?***

---

Using the test.zip file we discussed in the previous question, we could extract the entire contents of the temp subdirectory easily:

```
pkzip25 -extract -directories test.zip temp/*
```

If we did it as shown above we would not only extract all the files in the "temp" subdirectory, but also the "tut" subdirectory below it and any files it contains.

To extract only the "temp" subdirectories contents, and nothing else, we must exclude those directories we do not wish to extract:

```
pkzip25 -extract -directories test.zip temp/* -exclude=temp/tut/*
```

If the "temp" subdirectory had multiple subdirectories nested in it, you would need to exclude each one individually on the command line.

### ***What is AV?***

---

AV is short for "Authenticity Verification". AV is a process whereby a copy of PKZIP has unique codes and information contained in it identifying the owner of the copy of PKZIP. This information is then encoded into a .ZIP file when it is created. When the file is extracted by PKZIP, this information is checked. If the .ZIP file has been modified by a copy of PKZIP other than the one that initially created it, PKZIP will report that the file has been tampered with. The advantage of this feature is to offer a layer of protection between the creator of an archive and the recipient. The recipient knows that the file received is the file that was sent, as well as being able to identify the creator if it is not known.

**Note:** PKZIP can *only* authenticate AV information encoded by other versions of PKZIP (e.g. PKZIP for DOS 2.04g). It *cannot* encode AV information into .ZIP files.

### ***I forgot my password; what do I do?***

---

1. Try to remember the password.
2. Try passwords that are "close" to what you think it was.
3. Try mixed upper and lower case versions of your password.

**Don't forget or lose your passwords!** PKWARE has no special means for "getting around" the encryption and may not be able to assist in the recovery of an encrypted file. To help avoid the loss of data, you may wish to keep a written copy of your password(s) in a secure place.

### ***What does "Unknown Compression Method" mean?***

---

There are many different methods of compression. In the history of PKZIP alone, there have been seven different methods to date. The .ZIP file format was designed so that additional methods of compression can be added as they are developed. Therefore, the .ZIP file format will never need to be abandoned. This means that the .ZIP file in question was created or updated by a newer version of PKZIP than is being used to extract the data. You must use a newer version of PKZIP to extract these files.

### ***How can I make PKZIP run faster?***

---

PKZIP defaults to a compression method that is average in both compression amount and speed. If you want to get the most speed out of PKZIP, try the following:

1. Specify a faster compression method with a level sub-option (e.g. `-level=0`) or with a speed compression option alone (e.g. `-speed`). See page 73 for more information.
2. Compression speeds are highly dependent on the location of files being added, as well as the temporary file PKZIP creates when performing certain compression operations. If these files are located on a network drive, you may want to move them to a local drive before running PKZIP. Be aware of the effects file location can have on PKZIP's speed.

### ***How many files can be in a .ZIP file?***

---

There is no limit to the number of files you can add to a .ZIP file. However, if you use the **204** option for PKZIP 204g compatibility, your .ZIP file may contain no more than 16,383 file entries.

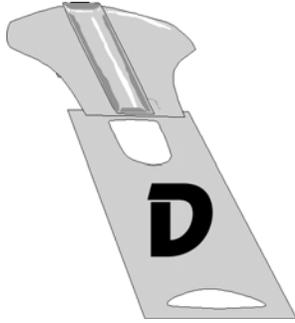
### ***Can I send a .ZIP file to a different type of computer?***

---

As of the publication of this manual, PKWARE supports PKZIP on MS-DOS, Windows(3.x, 95, NT(x86, DEC Alpha, PowerPC), OpenVMS, Sun Solaris, and OS/2 platforms. PKWARE intends to support additional platforms and will announce this support as it becomes available.

Because PKWARE has dedicated the .ZIP file format to the public domain, it is possible for other people to write programs which can read .ZIP files. We are aware of PKZIP compatible programs for a number of different platforms. A .ZIP file can be transferred to any platform for which you can find a compatible extraction program. Extraction and Compression programs not developed by PKWARE may not be completely compatible with the .ZIP file standard. Contact PKWARE for a list of platforms for which PKZIP and PKZIP compatible software is available.





## Appendix D

### How Does PKZIP Work

This Appendix provides a description of how PKZIP actually does its job. It isn't necessary for you to know or understand the information presented here, any more than you need to know how your carburetor works to drive a car. It is presented to help you feel more knowledgeable about the software.

#### Two Processes

PKZIP performs two functions: compression and archiving. Although the two ideas may seem related, they are actually completely separate.

- **Compression** is the process of representing a given piece of information with a smaller piece of information.
- **Archiving** is the process of combining many files into a single unit, usually along with vital information about each file.

#### Compression

The actual process used by PKZIP for its compression is too complex to explain in detail. Instead, some of the general principles behind information theory and compression are explained.

In order to understand data compression, you need to understand two ideas: Information Content and Binary Coding.

## Information Content

Everything in your computer, everything you ever read, is "information". The more complex a message is, the higher the information content. The less complex, the less "random" a message is, the lower the information content.

If a message contains a low amount of information, it should be possible to represent it in a smaller amount of space. Look at this page, for example. How much of the page is white space with no letters (information) on it? If you took away all of the white space this page would be significantly smaller. How many times are the words "the", "information" and "compression" on this page? If you could replace each of these words with something smaller, you would save a significant amount of space.

The more frequently the same group of symbols (in this case, letters) appear, the lower the information content of the message.

The "Field of Information Theory" uses the term *entropy* to describe the "true" information content of a message. Formulas can be used to determine the *entropy* of a message. The idea behind data compression is to derive a new smaller message from a larger original message, while maintaining the *entropy* of the original message.

As a simple example, consider this sentence:

she sells sea shells by the sea shore

This sentence is 37 characters long, including spaces. The spaces cannot be simply thrown away as the meaning of the original message would be lost.

There are obvious patterns to the sentence. The combination 'se' appears three times, 'sh' three times, and 'lls' twice. In fact, the 'se' pairs all have a space in front of them, so these can be ' se'.

she sells sea shells by the sea shore

We can replace each of these patterns with a single character:

#=" se"

\$="sh"

%="lls"

Note that the first replacement string includes a space at the beginning. If we reproduce the sentence with these symbols, it now looks like:

\$e#%#a \$e% by the#a \$ore

The new representation is 24 characters long; this is a saving of 13 characters, or 36%.

### **Binary Data Representation**

All information used, stored, and processed by computers is represented by two values, *zero* and *one*. Everything that you see on your screen, everything stored on disk, is represented by combinations of zero and one.

You can think of it as a sort of Morse Code. In Morse Code there are also only two values, dot and dash. When a computer stores a character, it uses a combination of eight zeros and ones.

Having eight positions in which to store a zero or one gives the computer 256 different possible combinations. You arrive at this number of combinations in this way:

If you have one coin, it can be in either of two positions: Heads(0) or Tails(1)

0 or 1

If you have two coins, there are four possible combinations:

00, 11, 10, 01

If you have three coins, there are eight possible combinations:

000, 001, 010, 011, 100, 101, 110, 111

As you can see, each time you add another coin (binary digit), the number of possible combinations doubles: 2, 4, 8, 16, 32, 64, 128, 256.

The computer uses eight binary digits to get 256 possible values. These values are mapped onto a table called ASCII (**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange). Each different combination has a particular character that is mapped to it, such as a letter, number or symbol. Each of these positions of 0 or 1 is called a **bit**.

she sells sea shells by the sea shore

The sample message above would be represented by 296 bits (37x8 bits).

If we follow standard ASCII, we have 256 different symbols being represented for our use. The sample sentence we are using only contains alphabetical characters, and only 11 of them at that. If we only need 11 different values, we could use a lower number of bits per character.

The closest value to 11 using binary combinations is 16 combinations, using 4 bits per character. If we wrote a new table of our own using four bits per character, and used it to represent the message, we would use only 98 bits. This would be half as many bits, a considerable savings.

We can do better!

It is possible to have binary codes of varying length. To do this we must use codes with unique values that are not repeated as the beginning of another code. In this way, we can find the codes in a long stream of zeros and ones.

If the codes were not constructed to have unique beginnings, it would not be possible to find each individual code within a long stream of zeroes and ones.

There are many types of coding techniques that produce codes of varying length, based upon symbol frequency. Some well-known coding schemes are Huffman and Shannon-Fanno. PKZIP uses Huffman encoding. The scheme is too complex to document here fully, however, we will discuss some rudiments of encoding. It is necessary for you to understand the principles described here.

A table of variable length codes for 11 symbols would look like this:

11	1101
110	0100
101	1000
001	01010
1011	00000
0010	

As you can see, the codes are getting longer and longer. Because of this, we will get the best results if we map the shortest code to the most common symbol in the message. If you know Morse code, or have occasion to look at it, you will notice that frequent characters, such as 'e', 't', 's' and so on have shorter codes assigned to them. Morse code tends to be about 25% more efficient because of this than it would have been had the codes been assigned at random.

A useful idea here is to allow a symbol to be not only a character, but also a group of characters.

Using the common patterns found in the first analysis of the message, we can map the following table:

O	currences	Symbol	New Code	Bits i	Message
	4	e	11		8
	4	(space)	110		12
	3	'se'	001		9
	3	sh	101		9
	2	lls	1011		8
	2	a	0010		8
	1	b	1101		4
	1	y	0100		4
	1	t	1000		4
	1	o	01010		5
	1	r	00000		5

Our new coding scheme can represent the message with only 74 bits. This is a savings of 222 bits from the 296 bits used in the "natural" encoding. This is one quarter of the original message size.

One important factor that would affect a real situation is the table we are using. In order for the data to be re-created from the "compressed" representation, we must include a copy of the table used to encode the data.

This can be a seriously limiting factor. If the data is too complex, or the encoding scheme too inefficient, the table used can be as big as the space saved by the encoding. In the worst cases, an attempt to re-encode the message using a table results in the encoded message plus the table being larger than the original message!

This is why data which uses a low number of symbols and frequently repeated combinations of symbols, such as a text file, compresses well. Complex, highly random data, such as the information representing a program on disk is difficult to encode efficiently, and therefore compresses less.

### **Speed vs. Size**

Searching for these patterns, and determining an efficient way to encode the data, takes a lot of computer power and time. The more time taken to analyze the data the better the compression will be. To get more speed, you must sacrifice some level of compression.

There are other steps and methods used in powerful compression schemes such as those used by PKWARE products. Hopefully this explanation gives you a better understanding of what happens when PKZIP compresses data.

### **Archiving**

Programs usually rely heavily on associated data files, or may actually consist of several related programs. Some programs may require dozens or even hundreds of files.

In the "dawn" of the PC age, people wanted a way to keep all of these associated files in one location. "Library" programs were created to take a number of files and group them together into a single file. This made them easier to find, easier to store, and much easier to send to someone by modem. It makes much more sense to be able to send someone a single "package" instead of many files. If you forget a file, all sorts of problems arise.

These programs were the birth of Archiving. In order for a single file to hold many files, information about each file also had to be stored in the archive. This information could then be used by the archival software to locate a file and pull it out, or to list information about the files contained within an archive.

Compression was first available as a utility that would take a single file and produce its compressed equivalent. People began to group files together with a Library program and then compress the archive file.

The next and obvious step in this process was to combine the two ideas. Compress the files and archive them. This made storage very simple; the compression was no longer a separate step and could be taken for granted as part of the archiving process.

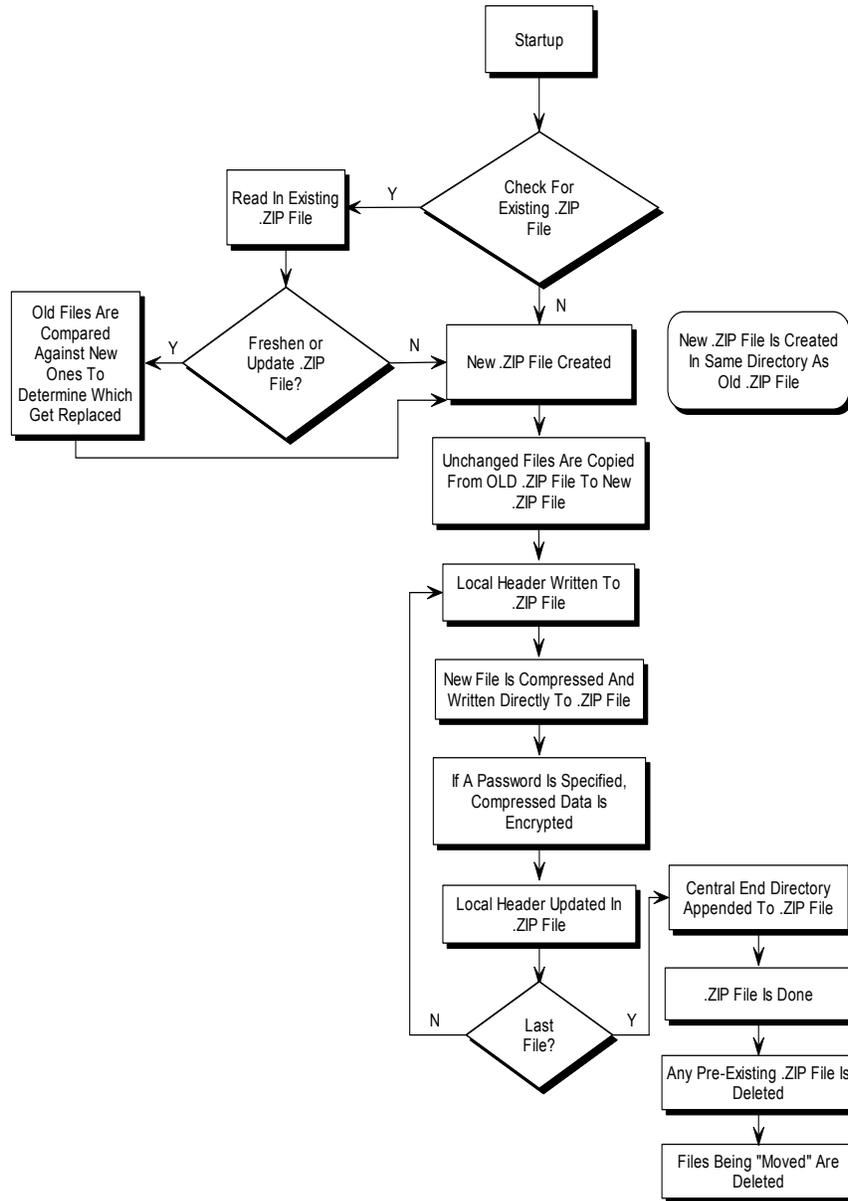
PKZIP is the second generation of these programs. PKZIP can not only compress and archive files, but also stores a great deal of vital information about the files. PKZIP even stores directory structures and volume labels.

## How PKZIP builds a .ZIP File

When you specify a PKZIP command line, PKZIP goes through several steps:

1. Parsing the command line.
2. Reserves the memory it will need to perform the compression, archiving and buffering.
3. Next, PKZIP looks for a .ZIP file with the same name as the one you specified on the command line. If it finds one, PKZIP reads the information on the files that it contains.
4. PKZIP then performs the requested action; it builds a new .ZIP file if none was found.
5. PKZIP reads the information from the command line specifying what files it is supposed to take, what files it shouldn't take, and if there is an **exclude** command.

## How PKZIP Builds A .ZIP File



- If a @list file is used, PKZIP reads it, then checks for which files exist. If a pattern is specified in the @list file, PKZIP generates a list of the files which match this pattern.
- If directory recursion has been specified with the **recurse** option, PKZIP next looks for any subdirectories. If it locates subdirectories it goes into them and looks for any files matching the files specified on the command line or in the @list file. If PKZIP finds subdirectories in the subdirectories, it repeats the process. It will continue this process until it finds no additional subdirectories.

Now PKZIP has a list in memory of all the files it should take. The files specified for exclusion are now compared against this list, and any that match are removed. If after this step is complete the list in memory is empty, PKZIP finishes with a message "Nothing to do!".

Now PKZIP reads-in each file, one at a time, and compresses it. When it is finished compressing a file, it adds it to the .ZIP file being created.

6. As PKZIP reads each file, it computes a CRC value for it. This CRC value is stored as part of the information concerning the file.

## CRC

This is an acronym for Cyclic Redundancy Check. When a CRC is performed, the data making up a file is passed through an algorithm. The algorithm computes a value based upon the contents resulting in an eight digit hexadecimal number representing the value of the file.

If even a single bit of a file is altered, and the CRC is performed again, the resulting CRC value will be different. By using a CRC value, it can be determined that there is an exact match for a particular file.

PKZIP calculates a CRC value for the original file before it is compressed. This value is then stored with a file in the .ZIP file. When a file is extracted it calculates a CRC value for the extracted data and compares it against the original CRC value. If the data has been damaged or altered, PKZIP can recognize and report this.

7. When PKZIP adds the compressed file to the .ZIP file, it first writes out a "Local Header" about the file. This Header contains useful information about the file, including:
  - The minimum version of PKZIP needed to extract this file
  - The compression method used on this file
  - File time
  - File date
  - The CRC value
  - The size of the compressed data
  - The uncompressed size of the file
  - The file name
  
8. After PKZIP has written all of the files to disk, it appends the "Central Directory" to the end of the .ZIP file. This Directory contains the same information as the Local Header for every file, as well as additional information. Some of this additional information includes:
  - The version of PKZIP that created the file
  - A comment about each file (if any)
  - File attributes (Hidden, Read Only, System)
  - Extended Attributes (If Specified)

## **Deleting Files From a .ZIP File**

PKZIP deletes files from a .ZIP file in the following manner:

1. PKZIP reads in the names of all the files contained in the .ZIP file.
2. PKZIP compares this list against the files you wish to delete.
3. Whatever files remain are moved into a new .ZIP file.
4. The original .ZIP file is superseded by a newer version of the .ZIP file.

This means that in order to delete files from a .ZIP file, you must have enough disk space to hold the existing .ZIP file as well as the .ZIP file that holds the deleted files.

## **Adding To an Existing .ZIP File**

Adding files to a .ZIP file is the same as creating a .ZIP file, but with one difference. Before PKZIP begins to add files, it first reads in the files that were in the existing .ZIP file. These old files and the new files are then both written out to a new .ZIP file, the old files being superseded by the new .ZIP files. This means that there must be enough free space for the old .ZIP file as well as the new .ZIP file to co-exist.



## Index

### —A—

Archive Attribute, 48  
archiving, 159, 165  
Archiving, 159, 165  
ASCII, 162  
Attributes  
    Storing, 59  
Authenticity Verification, 95  
AV, 95

### —B—

Binary, 159  
Binary Data Representation, 161

### —C—

Changing Defaults Using the  
    Configuration File Chapter, 107  
Clearing Archive Attributes, 47  
Combining Options, 37  
Command Characteristics Chapter, 113  
Command Line  
    Including an Option, 36  
Command/Option Character  
    Changing, 117  
Commands and Options, 35  
    Abbreviating, 37  
    Default Values, 44  
    Difference, 36  
    Values, 39  
Commands/Options  
    204, 65, 120  
    add, 45, 120  
    after, 55, 88, 120  
    attributes, 60, 121  
    before, 56, 88, 121  
    comment, 67, 122  
    configuration, 107, 122  
    console, 93, 122  
    default, 122  
    delete, 123

directories, 52, 92, 123  
exclude, 58, 90, 124  
extract, 85, 124  
fast, 74, 124  
filetype, 62, 125  
fix, 103, 126  
header, 71, 126  
help, 3, 126  
id, 126  
include, 56, 89, 127  
level, 73, 127  
license, 4, 127  
links, 127  
listchar, 116, 128  
listsfxtypes, 128  
locale, 114, 128  
mask, 65, 129  
maximum, 74, 129  
more, 130  
move, 77, 130  
namesfx, 130  
nametype, 90, 130  
noextended, 63, 130  
nofix, 131  
normal, 74, 131  
nozipextension, 131  
optionchar, 117, 131  
overwrite, 91, 132  
password, 69, 132  
path, 49, 133  
permission, 133  
preview, 102, 133  
print, 101, 134  
recurse, 48, 134  
sfx, 78, 134  
shortname, 59, 134  
silent, 135  
sort, 75, 93, 135  
span, 82, 136  
speed, 74, 136  
store, 74, 136  
temp, 104, 105, 136  
test, 101, 137  
times, 137  
version, 138

- view, 98, 138
- volume, 71, 138
- warning, 138
- zipdate, 72, 139
- Compressing
  - All Files in a Directory, 45
  - All Files in the Current Directory, 19
  - Files Based on Some Criteria, 55
  - Files From a Different Location, 20
  - Files in SubDirectories, 48
  - Files Newer Than a Specified Date, 55
  - Files Older Than a Specified Date, 56
  - Files That Match a Pattern, 18
  - Files With a LIST File, 54
  - New and Existing Files, 45
  - Only Files That Have Changed, 47
  - Only New Files, 46
  - Selected Files in the Current Directory, 17
  - Setting the Compression Level, 73
  - Single File in the Current Directory, 15
  - Specifying a Compression Level by Number, 73
  - Specifying a Compression Level by Option, 74
  - Specifying a Different Location for the .ZIP File, 21
  - Storing and Re-Creating Directory Path Information, 52
  - Storing Directory Path Information, 49
- Compressing Files
  - Based on File Type (UNIX), 62
  - Learning the Basics, 14
  - That Contain Certain Attributes, 60
- Compressing Files Chapter, 43
- Compressing Files to Diskette, 82
- Compression
  - Zipfile, 159
- CompuServe, 7
- Conventions Used in Tutorials, 13
- Converting a .ZIP file to a Self-Extracting file, 81
- CRC, 167, 168
- Create a Temporary .ZIP File On a Alternate Drive, 104
- Customer Support Form, 8

**-D-**

- Date and Time Environment Variables
  - Changing, 114
- Date of the .ZIP File, 72
- Default Value
  - Changing, 110
- Defaults
  - Resetting All Defaults, 112
  - Resetting Individual Defaults, 112
  - Resetting Original Defaults, 111
- Displaying a Brief View of a .ZIP File, 99
- Displaying a Detailed View of the .ZIP File, 100

**-E-**

- Electronic Mail - PKWARE, 6
- Entropy, 160
- Error and Warning Messages, 141
- Error Messages, 141
- Excluding Files From Being Compressed, 58
- Extended Attribute Storage, 63
- Extended Attributes and 204g Compatibility, 65
- Extracting
  - All Files From a .ZIP File, 31, 85
  - Excluding Files From Being, 90
  - Files Based on Some Criteria, 87
  - Files in a Specific Format, 59, 90
  - Files Newer Than a Specified Date, 88
  - Files Older Than a Specified Date, 88
  - Files Only for Display, 93
  - Files With a LIST File, 94
  - Including Files That Match a Pattern, 89
  - Multiple Files From a .ZIP File, 29
  - New and Existing Files, 85
  - Newer Versions of Existing Files and New Files, 86
  - Only Newer Versions of Files, 87
  - Overriding Default Files, 86
  - Retaining Directory Structure, 92
  - Selected Files, 30
  - Single File From a .ZIP File, 27
  - Sorting, 93

Extracting Files  
    Further Information, 35  
Extracting Files Chapter, 83  
Extracting Files To a Specified  
    Directory, 34  
Extracting Files: Learning the Basics, 26

**-F-**

Field Of Information Theory, 160  
File Naming Conventions, 14  
Fixing a Corrupt .ZIP File, 103  
Frequently Asked Questions, 151

**-G-**

Getting License Information, 4  
Getting Started Chapter, 11  
Getting Version Information, 5

**-H-**

Header Comment, 71  
How Does PKZIP Work, 159  
How PKZIP Builds a .ZIP File, 165

**-I-**

Including a Header Comment, 71  
Including a Password, 69  
Including a Text Comment, 67  
Including a Volume Label, 71  
Including Additional Information in a .ZIP  
    File, 67  
Including Files That Match a Pattern, 56

**-L-**

LIST Files, 54, 94  
    Changing the LIST Character, 116

**-M-**

Moving Files Into Your .ZIP File, 23  
Moving Files to a .ZIP File, 77

**-O-**

Overwriting Files, 91  
Overwriting Files That Already Exist in  
    Your Directory, 34

**-P-**

password, 155  
Passwords, 69  
Performing Miscellaneous Tasks on .ZIP  
    Files Chapter, 97  
PKWARE BBS, 7  
PKWARE Technical Support, 8  
PKZIP Options Reference, 119  
PKZIP, How Does It Work?, 159  
Preparing Your Workspace, 13  
Previewing Command and Option  
    Operations, 102  
Printing the Contents of a .ZIP File, 101

**-R-**

Removing File Attributes When  
    Compressing, 65

**-S-**

Self-Extracting File  
    Creating, 78  
Self-Extractor Options, 81  
Self-Extractors  
    Creating a Native Self-Extractor, 79  
    Creating DOS Junior, 80  
    Creating Regular DOS, 79  
    Creating Windows 16-Bit, 80  
    Differences Between Regular and  
        DOS Junior, 81  
Sorting Files Within a .ZIP File, 75  
Spanning, 82  
Storing File Attribute Information, 59  
subdirectories, 153, 154, 167  
Suppressing Screen Output, 105

**-T-**

Testing the Integrity of a .ZIP File, 101

**—U—**

UNIX Command Line, 3  
Using Help, 3  
Using PKWARE Technical Support, 6  
Using the Customer Support Form, 8  
Using the Tutorials, 12

**—V—**

Viewing Files Within a .ZIP File, 24

Viewing the Configuration File, 108  
Viewing the Contents of a .ZIP File, 98  
Volume Label, 71

**—W—**

Warning Messages, 146  
World Wide Web - PKWARE, 6