The Basics - UNIX

Welcome to PKZIP/SecureZIP Command Line. PKZIP/SecureZIP command line provides a command-line interface that enables you to access the functions of these two powerful data security and data archiving programs in scripts and batch files.

SecureZIP is an enhanced version of PKZIP. Both programs enable you to create and manage ZIP files and archives of other types, and both programs enable you to decrypt archives encrypted with either program. But SecureZIP provides additional features—most notably, commands and options for using digital certificates and OpenPGP keys to strongly encrypt and attach digital signatures to archives. SecureZIP Enterprise Edition adds the ability to create and manage digital certificates and OpenPGP keys.

This chapter will get you quickly up and running. After a brief overview of the manual and basic PKZIP concepts, you'll learn how to create ZIP archives and extract (unzip) files from archives. After covering the basic commands, you can get a taste of the power contained within PKZIP command options.

About This Manual

This manual describes the command-line features of all editions of PKZIP and SecureZIP for Windows, Mac OS X and UNIX/Linux.

In general, references to PKZIP in the text apply equally to SecureZIP. SecureZIP includes all the features of PKZIP. If a feature is available only with SecureZIP (and not with PKZIP) or requires the Enterprise edition of one of these programs, this is noted in the text.

The chapters group related commands and options and describe how to use them. This chapter provides an overview of basic program features. See the "Understanding Commands and Options" section for an explanation of how commands and options work.

You can customize the default behavior of most commands and options. "Changing Defaults for Commands and Options" describes how.

The Command Reference page contains a complete reference to the commands and options of the program. Experienced users may find that this appendix contains most of the information they need.

Conventions in This Guide

Most commands and options discussed in the following chapters work on all platforms that PKZIP supports. The cases are noted where a command or option is specific to a platform or operating system.

The name of a command or option appears by itself in ***bold italic*** font immediately under the main heading of the section where the command or option is discussed. In sections devoted to a particular sub-option, or value, of a command or option, the command or option is followed by an *equals* sign (=) and the name of the sub-option—for example, ***extract=all***.

An Overview of What PKZIP Does

PKZIP was developed to handle two basic tasks: It collects (adds) files into a container called an archive, and it pulls out (extracts) files from archives to restore them to their original state. The PKZIP ***add*** command is used to add files, and the ***extract*** command extracts them. These are the two most important PKZIP commands.

When PKZIP adds files to a specified archive, it creates the archive if it does not already exist. Generally, PKZIP compresses the added files so that they take less space, and it can also encrypt them so that they cannot be read by anyone who lacks the means to decrypt them.

As the creator of an archive, you control how its files are to be decrypted and by whom. You can encrypt files using a passphrase, such that the passphrase is required to decrypt them, or, if you have SecureZIP, you can use digital certificates to encrypt them such that only designated recipients can decrypt. SecureZIP also enables you to digitally sign files that you add to an archive, and the archive itself. A digital signature assures that the files really come from you and that it has not been unknowingly altered by someone else.

Compression, encryption, and signing are done when you add files. When you extract files, PKZIP decrypts the files, decompresses them, and validates any digital signatures.

Most PKZIP options relate to the two main operations of adding and extracting files and are for optional use when you do one of those things. For example, besides the options to encrypt or sign files, there are options for picking the files that you want to compress or encrypt and options for how you want to compress or encrypt them. Commands are also available for managing archives—for example, for testing their integrity and viewing their contents.

Supported Archive Types

An *archive* is a kind of file that can contain other files. Several types of archive files exist. Some can contain only one file, some can contain multiple files, and there can be other differences as well. A ZIP archive can contain multiple compressed files. ZIP files can be opened on almost every computing platform there is; this is the kind of archive that PKZIP creates by default and is the kind that you will probably use most often. Encryption and digital signing are supported only for ZIP and OpenPGP archives.

PKZIP enables you to create and extract from many other archive types besides ZIP. You do not need to do anything special to use PKZIP with one of these other archive types. PKZIP can tell what type an archive is and will just go ahead and extract its files. If you want to create a new, non-ZIP archive, there are two ways to tell PKZIP what type of archive to create:

- Specify a name for the archive file that uses the file name extension commonly associated with that archive type
- Use the ***archivetype*** option to specify the type of archive that you want

The following table lists the types of archives that PKZIP can create or extract from and the file name extensions customarily associated with these types. For some archive types, PKZIP can do extractions but cannot create new archives of that type.

Archive type	***PKZIP can create/extract***	***Usual file name extensions***
7Zip	Extract only	.7z
ARJ	Extract only	.arj
BinHex	Extract only	.hqx
BZIP2	Create and extract	.bz2
CDR	Extract only	.cdr
compress (UNIX, LZW)	Extract only	.Z
GZIP	Create and extract	.gz
IMG	Extract only	.img
ISO	Extract only	.iso
JAR	Create and extract	.jar, .ear, .war
LZH	Extract only	.lzh
OpenPGP	Create and extract	.pgp, .gpg
RAR	Extract only	.rar
TAR	Create and extract	.tar
UUEncoded	Create and extract	.uue
XXEncoded	Create and extract	.xxe
ZIP	Create and extract	.zip, .zipx

Your Work Environment: The Command Line

Your work area is a character-based command line, or shell. Type a command, and press **Enter** to execute the command. UNIX/Linux users can choose from a variety of shell environments and every graphical environment has a terminal or console app to execute PKZIP/SecureZIP commands and scripts.

Entering Commands

The syntax for commands entered on the command line is shown below. Brackets set off elements that are optional (Do not type the brackets.). Note that both PKZIP and SecureZIP Command Line use the same program name, ***pkzipc***, as shown below.

```
pkzipc [command] [options] zipfile [@list] [files...]
```

Examples:

To do this	***Command line***
Add specified files to an archive	***pkzipc -add zipfile.zip addfile.txt addfile2.doc***
Add to an archive all files in current directory	***pkzipc -add zipfile.zip*** or: ***pkzipc -add zipfile.zip ****
Add to an archive all files in a specified directory	***pkzipc -add zipfile.zip subdir****
Add files with the fast compression option	***pkzipc -add -fast zipfile.zip***
View list of files in archive	***pkzipc zipfile.zip***

View list of files whose names begin with "f" in archive	***pkzipc zipfile.zip f****
Extract all files from an archive	***pkzipc -extract zipfile.zip***
Extract specified files from an archive	***pkzipc -extract zipfile.zip readme.txt mystuff.doc***

A PKZIP command line has these main elements:

- **The name of the program executable**—***pkzipc***. This command runs PKZIP and must appear first.
- **A PKZIP command** for the main task you want PKZIP to do—for example, add files to an archive. Precede the command with a hyphen: -add
- **Any PKZIP options** that you want to use. For example, when adding files to an archive, you can use the ***maximum*** option to have PKZIP take a little extra time to compress them as much as possible. You can include zero or more options. Precede each with a hyphen: -maximum
- **The name of an archive file**, such as a ZIP file, to create or operate on.
- **The names of files** to operate on—for example, to add to an archive, to act on a file in an archive (for example, to delete it), or to extract from an archive. Alternatively, you can give a file name pattern such as *.doc to specify these files, or the name of a file that contains a list of such files.

    The name of the archive file must precede any other file names or file name patterns.

    To reference multiple file names and/or patterns to operate on, separate the names with spaces.

- The pathname of a destination folder to extract to. PKZIP extracts to the current folder by default. To extract to a different folder, specify the folder's pathname.

Note: When identifying a pathname that includes a space, always put the pathname in quotation marks. For example, if you are archiving all the files in the *Important Documents* directory, type the following:

> ***pkzipc -add zipfile.zip "Important Documents*"***

The only elements that are required in any command line are the name of the executable ***pkzipc*** and a PKZIP command. Other elements may be required depending on the commands or options used.

The order of appearance of the elements is not important except that:

- ***pkzipc*** must appear at the beginning of the command line
- The name of an archive file, if given, must appear before the name of any other file or folder

Creating a New Archive and Adding Files

Use the ***add*** command to add files to a new or existing archive.

For example, to add a file called test.txt to an archive file called temp.zip, use a command line like the following:

> ***pkzipc -add temp.zip test.txt***

If the archive does not already exist, PKZIP creates it.

You can optionally encrypt files when you add them. See "Encrypting Files That You Add to an Archive."

The following sections describe several ways to add files and how to display a listing of the files an archive contains.

Archive File Naming Conventions

Conventionally, archive files are named with a file name extension (the last part of the name, after the dot) that indicates the kind of archive. Thus a .ZIP archive generally has a name of the form myarchive.zip, where the file name extension is .zip. A BZIP2 archive generally has a file name extension of .bz2. See a list of archive type extensions in "Supported Archive Types" in this chapter.

PKZIP can both create and extract from a variety of archive types—including BZIP2. Because the file name extension is generally a good guide to the type of archive, PKZIP can use this information to determine what sort of archive you want to create. Here are the rules PKZIP uses to determine the type of archive to create:

- If you specify an archive name with an extension—for example, `myarchive.zip` or `myarchive.bz2`, PKZIP creates an archive of that name. Also, by default, PKZIP uses the file extension to select the type of compression to use. For example,

    pkzipc -add myarchive.zip

    results in a ZIP-format archive containing files compressed using standard ZIP-style compression (that is, using the Deflate compression algorithm). Alternatively, the following command line creates a BZIP2 archive. A BZIP2 archive is created using the BZIP2 compression algorithm and can contain only a single file.

    pkzipc -add myarchive.bz2 myfile.doc

- If you specify an archive name with *no* file extension, by default PKZIP creates a ZIP archive and adds a *.zip* extension to its name. For example:

    pkzipc -add myarchive

    produces a ZIP archive called `myarchive.zip`.

Note: The ***archivetype*** option lets you explicitly tell PKZIP the type of archive you want to create. See "Compressing Files to a Specified Type of Archive."

To suppress automatic adding of a file name extension on UNIX systems, use the ***noarchiveextension*** option.

Adding a Single File

To add a single file to an archive, use the ***add*** command and list on the command line the name of the archive and the name of the file to add. For example:

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

The command line adds file `red.txt`, in the current directory, to archive test.zip. Archive `test.zip` is created (in the current directory) if it does not already exist, or it is updated if it does exist.

The original of the added file `red.txt` remains in the current directory. Adding a file to an archive only compresses and adds a *copy* (unless you use the ***move*** option to delete the original).

Adding Multiple Files

You can specify multiple files to add either by explicitly naming the files or by using wildcard characters in a file name pattern.

Specifying Multiple Files by Name

To specify multiple files by name, list them on the command line, separated by spaces, after the name of the archive:

```
pkzipc -add test.zip green.doc blue.fil purple.txt
```

Specifying File Names that Match a Pattern

You can use file name patterns to specify, for example, all files whose names begin with p, or all .txt files. A file name pattern picks out all files whose names match the pattern.

Use these wildcard characters in file name patterns:

Wildcard character	*Matches*
Asterisk [*]	Zero or more characters
Question mark ?	Zero or one single character

For example, the following command line adds all files that have a particular file name extension (such as .txt):

```
pkzipc -add test.zip *.txt *.doc
```

The pattern **.htm?* in the command line below matches all files that end in .htm or .html:

```
pkzipc -add test.zip *.htm?
```

Consult the documentation for your operating system to learn more about using wildcards.

Adding All Files in the Current Directory

If you want to add all files in the current directory, you do not need to specify any files to add. Just use the ***add*** command with the name of the target archive:

```
pkzipc -add test.zip
```

This shorthand works only for adding all files in the current directory. To add all files in some other directory, you must use wildcards (or specify the files).

For example, both of the following command lines do the same thing: they add all files in the samples directory:

```
pkzipc -add test.zip samples*
pkzipc -add test.zip samples*.*
```

Adding All Files in a Different Directory

To add files in a directory other than the current directory, specify the path to the files. You can use either an absolute path or a path relative to the current directory.

For example, these command lines use an absolute path to specify files to add:

pkzipc -add test.zip /home/john_d/sales_reports/*.xls
pkzipc -add test.zip "/home/john_d/Documents/work samples/*.txt"

Enclose the path in quotes, as shown above, if it contains spaces. On UNIX, use a slash / instead of a backslash \ to indicate a subdirectory or set of files.

These command lines use a relative path to specify files to add:

pkzipc -add test.zip samples/sales_reports/*.xls
pkzipc -add test.zip ../records/jobs/*.doc

Working with an Archive in a Different Directory

If the target archive is not in the current directory, specify its location in the same way that you specify the location of files to add: include the path in the command line. You can use either an absolute or relative path.

pkzipc -add /usr/sales_reports/test.zip *.xls
pkzipc -add samples/test.zip sales_reports/*.xls

PKZIP still assumes that a relative path to files to add starts from the current directory even if the target archive is somewhere else. How you specify the location of the files is not affected by the location of the archive.

If a path contains spaces, enclose it in quotes.

Moving Files into an Archive

Normally, after you add files to an archive, PKZIP leaves the original files on your hard drive. If you would like PKZIP to delete the original files after adding copies to an archive, you can include the ***move*** option in the command line when you add the files.

pkzipc -add -move confidential.zip sales*.xls

The ***move*** option is useful if you want to remove files that you no longer expect to use or if you do not want to leave behind unencrypted copies of files that you have placed in an encrypted archive.

CAUTION: Be sure to keep backups of your important files. If you move your only copy of a file into an archive, and the archive becomes lost or damaged, you may be unable to recover your file.

For information on working with PKZIP options, see the section "Understanding Commands and Options" later in this chapter.

Viewing Files in an Archive

The ***view*** command produces a list of the files in an archive and various pieces of information about the files. Use the command to verify that files were added as expected or simply to find out what files an archive contains. It is also useful to see what path information is saved with a file. Path information is saved as part of the file name and so must be taken into account when you reference the file to extract it.

pkzipc -view myfiles.zip

The display generated by the ***view*** command looks like this:

```
Length Method   Size Ratio     Date       Time      CRC-32  Attr   Name
------ ------   ---- -----     ----       ----      ------  ----   ----
    0B Stored   0B     0.0%    4/4/2014   7:25p   00000000  ---wD  orderStatus_fi
 les/
 3557B DeflatN 3496B   1.8%    4/4/2014   7:24p   23ce6c93  -a-w-  orderStatus_fi
 les/bw_logo.gif
 1653B DeflatN  847B  48.8%    2/9/2014  11:06a   891d9c90  -a-w-  caroline.txt
 71B   DeflatN   66B   7.1%   1/27/2014  11:41a   fa66929c  -a-w-  dummy_list.txt
 420B  DeflatN  128B  69.6%   3/10/2014   6:23p   4b63fc2a  -a-w-  filelist.txt
 420B  DeflatN  128B  69.6%   3/10/2014   6:23p   4b63fc2a  -a-w-  filelist2.txt
 420B  DeflatN  128B  69.6%   3/10/2014   6:23p   4b63fc2a  -a-w-  filelist3.txt
 420B  DeflatN  128B  69.6%   3/10/2014   6:23p   4b63fc2a  -a-w-  filelist4.txt
 308B  DeflatN  122B  60.4%   5/10/2014   3:14p   5f177b65  -a-w-  files.txt
 24B   DeflatN   16B  33.4%   1/24/2014   2:27p   f22154bb  -a-w-  mylist.txt
 7915B DeflatN 1701B  78.6%  10/27/2014  12:08p   7b38176a  -a-w-  shared.txt
 1463B DeflatN  816B  44.3%    1/9/2014   6:54p   2ef75758  -a-w-  verisign.txt
 878B  DeflatN  432B  50.8%   8/26/2014  10:40a   d1c700e7  -a-w-  What's New.txt
 ------         ----  -----                                        ----
 17KB          8008B  54.4%                                        13
```

The listing above was generated from a Windows command line. On UNIX, the *Attr* column is replaced by a *Mode* column with permission numbers for each file.

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

See "Adding Files to an Archive in UNIX" for information on other options you can use when adding files, including options to set the level of compression, add encryption, and so on.

Extracting Files from an Archive

To get a copy of a file out of an archive in its original form so that you can use it again, use the extract command. Extracting decrypts the file if it was encrypted, decompresses it, and validates any digital signature attached when the file was added.

You can extract all the files in an archive, or just selected files. As with adding files, PKZIP gives you numerous options for picking files and for choosing how to extract them. See "Extracting Files."

Extracting All Files

To extract *all* files in an archive, include in the command line just the ***extract*** command and the name of the archive.

```
pkzipc -extract temp.zip
```

The files are extracted to the current directory.

Extracting Some Files

To extract only a selection of files, additionally specify the files to extract. For example, the following command line extracts all .txt files in the archive into the current directory.

```
pkzipc -extract temp.zip *.txt
```

You can also extract multiple files by explicitly listing their pathnames, separated by a space:

```
pkzipc -extract temp.zip green.doc blue.fil purple.txt
```

How you identify files in an archive depends on the path information that was archived with them. In an archive, path information is treated as part of a file name for purposes of identification. (Use the ***view*** command to see any path information saved with files.) For example, if you want to extract file `august.xls`, and the pathname of the file in the archive is records\august.xls, either of the following command lines will extract the file. The command line that contains the * wildcard character also extracts all other .xls files whose pathnames start with r.

```
pkzipc -extract temp.zip records\august.xls

pkzipc -extract temp.zip r*.xls
```

Extracting Files to a Different Directory

By default, files are extracted to the current directory. To extract files to a different location, specify a path. For example, the following command line uses the two-dots (..) notation to specify a path to the parent of the current directory, one level up.

```
pkzipc -extract temp.zip *.txt ..
```

A destination pathname can occur in the command line anywhere after (to the right of) the name of the archive. For example, the following command line extracts all files in data.zip to the january subdirectory of the current directory:

```
pkzipc -extract data.zip january
```

To create a january subdirectory if one does not already exist, append a slash:

```
pkzipc -extract data.zip january/
```

A folder name can appear before or after names of files to be extracted. Both of the following command lines extract *report.xls* to january:

```
pkzipc -extract data.zip report.xls january

pkzipc -extract data.zip january report.xls
```

PKZIP evaluates file or folder possibilities in the order they appear, from left to right, after the name of the archive. The first one found that is the name of a folder determines the destination folder.

Extracting New and Newer Files

By default, the ***extract*** command extracts all files if you do not specify particular files. You can also configure the ***extract*** command to extract only files that are newer versions of files already in the target directory, or only files that are newer versions or do not already exist in the directory.

For example, the following command line uses the ***update*** sub-option of the ***extract*** command to tell PKZIP to extract only files that are newer versions or do not already exist in the directory:

pkzipc -extract=update temp.zip

Sub-options are explained in the section "Commands and Options with Values," later in this chapter.

Using Filters When Selecting Files

You can use various criteria to identify a specified set of files to add or extract, so that you only select the subset of files that meets the filter criterion.

For example, the command line below specifies all text files to add, but uses the filter option ***after*** to add a constraint; namely, that a file must also have been modified after the specified date (*mmddyyyy*). As a result, only those text files that meet the additional requirement imposed by the ***after*** option are added.

pkzipc -add -after=03152006 03152011 myfiles.zip *.txt

All the filter options described in this section work with both ***add*** and ***extract*** commands.

Selecting Files by Date

before, after

The ***before*** option selects files that were modified before a specified date. The ***after*** option selects files that were modified on or after a specified date.

In the United States, enter dates in one of 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 9.

The following sample command line adds files dated before February 24, 2011:

pkzipc -add -before=02242011 test.zip

The command line below adds files dated February 24, 2011, or later:

pkzipc -add -after=02242011 test.zip

Selecting Files by Age

older, newer

The ***older*** and ***newer*** options select files that are older or newer than a specified age. You can list the age in days (the default), hours, minutes, or seconds using the abbreviations shown in the following table.

Time unit	*Abbreviation*
Days (default)	d (or nothing)
Hours	h
Minutes	m
Seconds	s

For example, the following command lines each add files that are no more than five days old:

pkzipc -add -newer=5 test.zip *

pkzipc -add -newer=5d test.zip *

The command lines below add files that are older than five days:

pkzipc -add -older=5 test.zip *

pkzipc -add -older=5d test.zip *

The following command line uses both options to select files to extract:

pkzipc -extract -newer=10 -older=5 test.zip *

With a time unit of days, the interval (for example, five days) is measured from the beginning of the current day. So, for example, if it is currently 3:34 p.m. on June 15, setting ***newer*** or ***older*** to 5 sets the cutoff to 12:00 a.m. June 10. The ***older*** option gets files dated earlier than this; the ***newer*** option gets files dated on or after this.

With time units of hours, minutes, or seconds, the interval is measured from the current system time. So, for example, the following command line selects files modified within the last 48 hours:

```
pkzipc -add -newer=48h test.zip *
```

Selecting Files by Size

larger, smaller

The ***larger*** and ***smaller*** options select files that are larger than or equal to, or smaller than or equal to, a size specified in bytes.

The following command line adds files whose size is in the range 5000-7000 bytes, inclusive:

```
pkzipc -add -larger=5000 -smaller=7000 test.zip
```

You can also use ***k***, ***m***, ***g***, or ***t*** to specify kilobytes (1024 bytes), megabytes (1024 Kb), gigabytes (1024 Mb), and terabytes (1024 Gb), respectively.

The following command line adds files whose size is in the range 500 megabytes to 1 gigabyte, inclusive:

```
pkzipc -add -larger=500m -smaller=1g test.zip
```

Selecting Files to Include or Exclude

include

The ***include*** option has two uses:

- To specify a file name pattern to use by default when selecting files to add or extract
- To override, in the current command line, a configured default setting that excludes files from being selected

Ordinarily, to select files whose names match a pattern (for example, *.doc), simply specify the pattern on the command line:

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

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

To include one or more file patterns automatically when selecting files, you can configure a default value for ***include***. For example, if you want to automatically include all files with the extension of .doc when adding files, enter the following:

```
pkzipc -config -add -include="*.doc"
```

This configured default causes a command line like the following to zip all .doc files in addition to the *.txt files explicitly specified.

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

You can also use ***include*** to override a default setting of the ***exclude*** option.

For example, if you have configured PKZIP to exclude *.txt files by default when adding, you can include such files in a particular case with the command line below:

```
pkzipc -add -include="*.txt" test.zip
```

If you do not need to override a default configuration setting, you do not need to specify the ***include*** option in your command: the file pattern by itself is enough.

For more information on modifying default configuration values, see "Changing Defaults for Commands and Options."

exclude

The ***exclude*** option has two uses:

- To specify a file name pattern or list file to use to exclude files by default when selecting files to add or extract
- To override, in the current command line, a configured default setting that includes files

To exclude one or more file patterns automatically when selecting files, you can configure a default value for ***exclude***. For example, if you want to automatically exclude all files with the extension of .doc when adding files, enter the following:

```
pkzipc -configuration -add -exclude="*.doc"
```

The command line below has the same effect but abbreviates the ***configuration*** option:

```
pkzipc -config -add -exclude="*.doc"
```

The configured default value for ***exclude*** causes a command line like the following to zip all files except .doc files.

```
pkzipc -add test.zip *.*
```

To exclude a list of files, specify the list file as the value of the ***exclude*** option:

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

You can also use ***exclude*** to override a default setting of the ***include*** option. For example, if you have configured PKZIP to include *.txt files by default, you can exclude them in a particular case with the command line below:

```
pkzipc -add -exclude="*.txt" test.zip
```

For more information on modifying default configuration values, see "Changing Defaults for Commands and Options."

Understanding Commands and Options

A PKZIP command line includes a command and can also include options that affect how the command is done or specify things to be done in conjunction with it. Many commands and options also have sub-options that determine how the command or option behaves.

Difference between a Command and Option

A command tells PKZIP *what* to do; an option tells PKZIP to do the main task in a particular way or to do some additional task in the course of doing the main task.

For example, the ***add*** command tells PKZIP to add files to an archive. You can use the ***maximum*** option with the ***add*** command to tell PKZIP to use maximum compression when adding the files. If you want to delete the original files after they are added, you can include the ***move*** option too:

```
pkzipc -add -maximum -move myarchive.zip *.doc
```

A command line must always contain a command; it can contain any number of options. A command stands alone in a command line, without requiring (or permitting) any other command. For this reason, it is sometimes referred to as a *standalone* to indicate that it is not an option. An option can be used only with a command.

A few options bend the rules in that they can be used either as options or as commands. These include ***comment***, ***header***, ***sfx***, ***sign***, and some of the ***mai*** options. For example, ***comment*** prompts you for a comment to attach to an archive. This option can be used with the ***add*** command to attach a comment to a new archive, or it can be used by itself to attach a comment to an archive that already exists.

Including an Option in Your Command Line

To use an option, prefix it with a hyphen and insert it in the PKZIP command line after the main command.

For example, the following command line uses the ***maximum*** option with the ***add*** command. This option tells PKZIP to use maximum compression:

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

The following example uses the ***overwrite*** option to turn off the usual prompting to overwrite files with the same names as files to be extracted. The command line directs that extracted files simply overwrite any files that have the same names, without prompting:

```
pkzipc -extract -overwrite test.zip
```

Abbreviating Commands and Options

In a command line, you can abbreviate commands and options by leaving off letters at the end as long as you give enough of the name for PKZIP to know what command or option you mean.

For example, you can abbreviate the name of the ***maximum*** option to ***max***, as in the command line below, because no other option name starts with those letters.

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

The command line below abbreviates the name of the ***extract*** command to ***ext***:

```
pkzipc -ext test.zip
```

Note: It's good practice to avoid abbreviating commands and options when writing scripts, as PKWARE adds new features with each new version. Using full commands ensures that your scripts will work regardless of what other commands may be introduced.

Using Multiple Options

To use multiple options in the same command line, separate them by spaces.

For example, the following command line includes both the ***maximum*** and ***comment*** options. These tell PKZIP to use maximum compression and to prompt you for a comment for each newly added file:

```
pkzipc -add -maximum -comment test.zip *.doc
```

The order in which options appear is not important.

Not all options can be used with all commands. For example, you cannot use ***maximum*** with the ***extract*** command. Appendix A lists the commands with which each option can be used.

Commands and Options with Values

Some commands and options have different possible values, called sub-options, that let you customize how the command or option behaves. For example, the ***level*** option enables you to specify how much compression you want to use (more compression takes longer). When you use ***level***, you specify a value for a particular level of compression. For example:

```
pkzipc -add -level=9 myarchive.zip
```

To specify a sub-option or value with a command or option, attach it to the command/option with an equal sign, as in the last example.

Commands as well as options can have sub-options. For example, you can use the ***add*** command to add all selected files to an archive, or to add only files that are newer versions of files that the archive already contains. You indicate how you want ***add*** to work by specifying a sub-option. To have the command add only newer versions of files that the archive already contains, use the command with the ***freshen*** sub-option:

```
pkzipc -add=freshen myarchive.zip *.*
```

Most commands and options that have multiple possible predefined values or sub-options use one of the values as a default. Some options are disabled by default, but if an option has a default value, that value is implicitly used in any command line that does not explicitly list the option.

For example, the ***level*** option has a default value of *5* (normal compression). The following command line does not explicitly include the ***level*** option, but because the option is not disabled and has a default value, the command line applies the option at its default value and uses normal compression:

```
pkzipc -add myarchive.zip *.*
```

PKZIP uses the default value for a *command* (as opposed to an option) whenever the command is used with no sub-option specified. In the preceding example, PKZIP uses the default value for ***add***.

You can replace original default settings with your own by using the ***configuration*** command. See "Changing Defaults for Commands and Options."

For a list of all commands and options together with their sub-options, see the Command Reference page.

Using Strong Encryption

PKZIP allows you to use either of two kinds of encryption to encrypt ZIP archives: the older, traditional PKZIP encryption, or strong encryption. Strong encryption is much more secure than traditional PKZIP encryption. Traditional encryption support is mainly intended for legacy content.

PKZIP and SecureZIP v14 added new support for encrypting and decrypting files using the strong OpenPGP (RFC 4880) standard. You can open and decrypt any OpenPGP files you receive with PKZIP and SecureZIP. Create OpenPGP-based archives and use its encryption on any file (not just ZIP archives) with SecureZIP in all environments, and with PKZIP in Windows.

Note: SecureZIP supports both X.509 digital certificates and OpenPGP key pairs. When this document refers to *certificate-based encryption*, or *recipient-based encryption*, you can use either type of key. To learn more about the differences between digital certificates and OpenPGP keys, see "Public-Key Infrastructure and Digital Certificates."

Traditional PKZIP encryption uses only passphrases. Strong encryption can be done with a passphrase (symmetric key), a public/private key pair (asymmetric key) or both. When you encrypt using a public/private key pair, only the owner of the private key—called a *recipient*—can decrypt. Using both a passphrase and recipient certificates widens the number of people who can open the encrypted file, both those on the recipient list with the proper private key and anyone with the passphrase.

Use the ***passphrase*** option to apply either traditional or strong passphrase-based encryption.

To do certificate-based strong encryption, use the ***recipient*** option to specify the owners of the certificates for whom you want to encrypt. You must also have a copy of each recipient's certificate that contains the certificate's public key.

With both certificate- and passphrase-based strong encryption, use the ***cryptalgorithm*** option to specify an encryption algorithm and key length (for example, AES, 256 bits).

You need version 6.0 or later of PKZIP (or ZIP Reader) to decrypt archives that were strongly encrypted using PKZIP. You may need SecureZIP to strongly encrypt archives yourself.To learn much more about encryption in PKZIP, see "Encrypting Files That You Add to an Archive," "Extracting Passphrase-Protected Files," and "Working with OpenPGP Files."