



PKZIP[®] 6.0
for the X Window System
Getting Started Manual

Copyright © 2002 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.

Portions of this software include RSA BSAFE® cryptographic or security protocol software from RSA Security Inc.

PKWARE, the PKWARE logo, the zipper logo, PKZIP, PKUNZIP, and PKSFX 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.

RSA and BSAFE are registered trademarks of RSA Security Inc.

Table of Contents

OPERATING SYSTEM REQUIREMENTS	5
INSTALLING AND CONFIGURING PKZIP	6
<i>Setting PKZIP in the Path</i>	6
PKZIPXCFG ENVIRONMENT VARIABLE AND PKZIPX.CFG FILE.....	8
<i>Editing the pkzipx.cfg File</i>	9
<i>Configuring Date Order in the pkzipx.cfg File</i>	10
<i>Configuring the Separator Character in the pkzipx.cfg File</i>	11
<i>Configuring Command Line Arguments Options in the</i>	
<i>pkzipx.cfg File</i>	11
<i>Sample pkzipx.cfg File</i>	11
AUTHENTICITY VERIFICATION.....	12
SETTING THE PKAVEXTRA ENVIRONMENT VARIABLE.....	15
DIGITAL CERTIFICATES	23
<i>Installing the ROOT, CA and SPC certificates:</i>	23
STRONG ENCRYPTION.....	26
FILE CAPACITIES	27
ENCODED AND ARCHIVE FILE SUPPORT	31

Operating System Requirements

Minimum System Requirements for PKZIP 6.0 for the X Window System.

Platform:	Minimum System Requirements:
Sun Solaris® 2.6 for SPARC	128 MB of RAM (256 MB recommended) UltraSPARC processor* CDE Installed (recommended)
IBM AIX® 4.3	64 MB of RAM (128 MB recommended) 200 MHz PPC processor† CDE/AIX Windows Installed (recommended)
HP-UX® 10.20 or greater	64 MB of RAM (128 MB recommended) 120 MHz 7200LC processor† CDE Installed (recommended)
Linux® running the 2.4 or later kernel	64 MB of RAM (128 MB recommended) 233 MHz Pentium II processor† LessTif 0.92.32 or later (required) glibc-2.2.4 (required)

* To run the program pkzipx, you must have an UltraSPARC processor. If you have an older SPARC processor, you should run pkzipx.v7 instead.

† We recommend that you use at least the listed processor and speed. Better processors speed up the application and your system. PKZIP works on slower processors with decreased performance.

Installing and Configuring PKZIP

Installing from a PKSFX file:

If you are installing a version of the software downloaded from the Internet you will receive a PKSFX self-extracting file. The Installer is packaged inside the PKSFX file. Run the PKSFX file to start the installation. It will extract the files needed to install the software and it automatically starts the setup program. Follow the instructions provided by the setup program to complete the installation.

We encourage you to make a backup of your PKSFX file, or of the contents of the intermediate folder, to ensure you will be able to reinstall this product if you ever need to in the future.

Setting PKZIP in the Path

To access PKZIP from any directory without typing a path, specify the PKZIP destination directory in your system's path environment variable. The path specification is typically located in your start-up file `.profile` or `.login`.

To add the PKZIP installation directory to your path:

1. Open your start-up file (e.g., `.profile` or `.login`) using a text editor such as `vi`, `Pico`, and `Emacs`.
2. What you do next depends on the shell you are using:

If you are using the Korn Shell (**ksh**) or the Bourne Shell (**sh**), look for a line similar to the one below in your `.profile` file.

```
PATH=/usr/bin:
```

If you are using the C Shell (**csh**), look for a line similar to the one below in your `.login` file:

```
setenv PATH /usr/bin:
```

or:

```
set path=(/usr/bin .)
```

A colon separates path designations in the PATH= and setenv PATH examples while a single space separates path designations in the set path= example.

3. Add the PKZIP installation directory (using a colon or space to separate as appropriate) to the PATH=, setenv PATH, or set path= lines. If, for example, you installed PKZIP in the /usr/local/bin/pkware/pkzipx directory, add the following:

```
/usr/local/bin/pkware/pkzipx
```

The path you specify depends on the location of the pkzipc file. If you use The C Shell, proceed to step 5. Korn and Bourne Shell users go to step 4.

4. Verify that the line below exists in your start-up file .profile or .login:

```
export PATH
```

If it does not exist, add it after the PATH= line.

5. Save and exit the file.
6. Reset your current environment settings. This can be accomplished by logging off your account. The next time you log on to your account, you can run pkzipc without specifying a path, regardless of the current working directory.

PKZIPXCFG Environment Variable and pkzipx.cfg File

To configure and customize commands and options in PKZIP, you can use the Options in PKZIP, or directly edit the configuration file (pkzipx.cfg) with a text editor. The recommended method for configuring and customizing PKZIP commands and options is via the Options under Tools. You may set the PKZIPXCFG environment variable if you do not want this file located in your home directory.

To set the PKZIPXCFG environment variable:

1. Open your start-up file .profile or .login using a text editor such as vi, Pico, and Emacs.
2. What you do next depends on the shell you are using:

If you use the Korn Shell (**ksh**) or the Bourne Shell (**sh**), add the following lines to your .profile file:

```
PKZIPXCFG=<path to the pkzipx.cfg file>
export PKZIPXCFG
```

If you use the C Shell (**csh**), add the following line to your .login file:

```
setenv PKZIPXCFG <path to the pkzipx.cfg file>
```

The path you specify for pkzipx.cfg depends on the location of the file. The pkzipx.cfg file typically resides in your home directory.

3. Save and exit the file.
4. To reset your current environment settings, log off your account. The PKZIPXCFG variable will be set the next time you log on to your account.

Editing the pkzipx.cfg File

You have the option of modifying the pkzipx.cfg file directly with a text editor. You can modify date formats, as well as command line argument options. The default pkzipx.cfg file used by PKZIP looks similar to the following:

```
# PKZip configuration file format section
:FORMATS
iDate=0
sDate=/
# PKZip configuration file option section
:OPTIONS
```

Entries in the pkzipx.cfg that follow a "#" character are treated as comments. Entries that follow the ":FORMATS" designation modify date format settings. Entries that follow the ":OPTIONS" designation modify command argument options.

Configuring Date Order in the pkzipx.cfg File

PKZIP uses MMDDYY as the default date order. You can specify the date order used by PKZIP if you modify the *idate* option after ":FORMATS".

The available *idate* sub-options are listed in the following table:

Sub-Option	To:	For example:
0	Set PKZIP date format to MMDDYY .	idate=0
1	Set PKZIP date format to DDMMYY .	idate=1
2	Set PKZIP date format to YYMMDD .	idate=2

Specifying *idate=1* in your *pkzipx.cfg* file, for example, sets the default PKZIP date format to DDMMYY. File information that appears in the *-view* option is presented in this format. Any files that you subsequently add or extract using filters require the use of this particular format in order for PKZIP to interpret it correctly.

Configuring the Separator Character in the pkzipx.cfg File

You can specify the field separator PKZIP uses by modifying the **sdate** option in the pkzipx.cfg. By default, PKZIP uses a "/" character to separate the date fields. If, for example, you want to change the field separator character from a "/" to a "-", you could add the following line to the pkzipx.cfg after the ":FORMATS" designation:

```
sdate=-
```

Specifying **sdate=-** in the pkzipx.cfg file, for example, sets the default PKZIP date format to MM - DD -YY. File information that appears via the *grid* is presented in this format. Any files that you subsequently add or extract using filters require this particular format to work properly. In all other places, you enter the date.

Note: The *idate* and *sdate* options are only configurable by editing the pkzipx.cfg file with a text editor.

Configuring Command Line Arguments Options in the pkzipx.cfg File

You may specify options in PKZIP by adding the appropriate command/option and sub-option (if applicable) after the ":OPTIONS" designation in the pkzipx.cfg. If, for example, you wish to set the options for all **add** operations to **update**, add the following line to the pkzipx.cfg file under the ":OPTIONS" designation:

```
-add=update
```

See the PKZIP Command Line User's Manual for options that are configurable.

Sample pkzipx.cfg File

Assume that you want to instruct PKZIP to use DDMYY as the default date order, as well as use the "-" character as the separator character. Additionally, assume that you want to set the **filetype** option to process all file types. To do this, edit the pkzipx.cfg file with a text editor. Specifically, modify the **idate** and **sdate** options under the ":FORMATS" designation and add the appropriate option under the ":OPTIONS" section. Your pkzipx.cfg file should look similar to the following example:

```
# PKZip configuration file format section
:FORMATS
idate=1
sdate=-
# PKZip configuration file option section
:OPTIONS
-filetype=all
```

Authenticity Verification

PKZIP allows you to embed an electronic signature with files stored in a .ZIP archive, and subsequently, authenticate those files on extraction. PKWARE Authenticity Verification (AV) information allows you to detect whether a .ZIP file's integrity has been compromised. You may use the **authenticity** command line option to instruct PKZIP to embed AV information with your .ZIP files. However, before you can use this option, you must first configure PKZIP for AV. To configure PKZIP for AV, complete the following steps:

1. Complete the Application for Authenticity Verification (authveri.txt) found in the PKZIP 6.0 installation directory (/usr/local/bin/pkware/pkzipx) or on page 17 of this manual. You may also access the Application for Authenticity Verification via the web at the following URL:

<http://www.pkware.com/authenticity>

2. Complete your application from our web site or fax (+1-414-354-8559) your completed application to PKWARE. PKWARE will process your application and

send you a confirmation message via the United States Postal Service or FAX containing two serial numbers. These serial numbers are required to configure PKZIP for Authenticity Verification.

3. After you receive your confirmation message, go to the command prompt and change to the PKZIP installation directory. Run the putav program by typing the following:

```
putavx
```

After you press **Enter**, you are prompted to enter your company name and the two serial numbers as they appear in the confirmation message you received from PKWARE. Screen output similar to the following appears:

```
This program will configure PKZIP for Authenticity
Verification. To proceed, complete and submit the Application
for Authenticity Verification located in the PKZIP installation
directory (e.g., authver1.txt) or in the PKZIP Getting Started
Manual. You may also access the Application for Authenticity
Verification via the world wide web at the following URL:
http://www.pkware.com/authenticity
```

```
-----
PKWARE will process your application and send you a
confirmation message via the United States Postal Service
or FAX. This confirmation message will contain two serial
numbers required to configure PKZIP for Authenticity
Verification.
```

```
-----
Press Ctrl-C to abort or return to continue
```

Press **Enter**. The following appears:

```
PUTAV Ver 4.0 - Put Authenticity Verification in PKZIP
Copyright 1990-2002 PKWARE, Inc. All rights reserved.
```

```
Enter company name exactly as it appears on the PKWARE documentation.
Company Name: Vandelay Industries
Enter serial numbers exactly as they appear on the PKWARE documentation.
Serial Number 1: 8SU214x
Serial Number 2: A7YG15w
```

```
working, done.
```

```
Serial number installation complete. Check value: 31f50e0d
```

You must enter your company name and the two serial numbers exactly as they appear in the confirmation message. The check value displayed on your screen should match the check value specified in your confirmation message. If they do not match, repeat step3.



You may include additional information such as telephone number and address with your AV string. To do this, place an ASCII text file containing this additional information named **AVEXTRA.TXT** in your current working directory, or the location specified by the PKAVEXTRA environment variable.

Setting the PKAVEXTRA Environment Variable

To include additional information with your AV string, you can place the text in a file and specify the file location in the PKAVEXTRA Environment variable.

You can specify the location of the pkavextra.txt file on your system with the help of an environment variable called PKAVEXTRA. To set the PKAVEXTRA environment variable, do the following:

1. Using a text editor such as vi, Pico, Emacs, open your start-up file .profile or .login.
2. What you do next depends on the shell you are using:

If you are using the Korn Shell (**ksh**) or the Bourne Shell (**sh**), add the following lines to your .profile file:

```
PKAVEXTRA=<path to the pkavextra.txt file>  
export PKAVEXTRA
```

If you are using the C Shell (**cs**h), add the following line to your .login file:

```
setenv PKAVEXTRA <path to the pkavextra.txt file>
```

The path you specify depends on the location of the pksfxs.txt file.

3. Save and exit the file.
4. To reset your current environment settings, log off your account. The PKAVEXTRA variable will be set the next time you log on to your account.

Digital Certificates

PKZIP allows you to digitally sign the individual files archived in a .ZIP file (as well as the Central Directory) and subsequently authenticate those files upon extraction. PKZIP signing functionality is based on the X.509 certificate standard and is therefore compatible with authenticity functionality in other applications such as Microsoft's Internet Explorer. Signing a .ZIP file allows you to detect whether a .ZIP file's integrity has been compromised. Before configuring PKZIP to sign files, you must first have a digital certificate to use for signing. These certificates must be a minimum of 1024-bit RSA format. Digital certificates are available from a variety of certificate authorities. Visit our web site for information on obtaining a certificate:

<http://www.pkware.com/aboutus/digicert.html>

Installing the ROOT, CA and SPC certificates:

The first thing you must do, if you want to verify signatures or sign .ZIP files is to install the ROOT and CA certificates. You might also want to include the SPC certificates, if any. To accomplish this, we suggest using the authenticode tools from Microsoft on a Windows computer with Internet Explorer 4 or better installed. If you do not already have these tools, Microsoft has them on their website.

If you have Internet Explorer 4 installed, visit the following URL:

<http://msdn.microsoft.com/MSDN-FILES/027/000/218/codesign.exe>

If you have Internet Explorer 5 installed, visit this URL instead:

<http://msdn.microsoft.com/MSDN-FILES/027/000/219/codesign.exe>

If you have IE 6, you will need to download Microsoft's ActiveX SDK to obtain the authenticode tools.

Unfortunately, no one has provided tools to obtain these certificates from other systems, such as Netscape running on your X Window system.

1. Run the following commands (on the windows computer):

```
certmgr -add -7 -all -s Root Root.p7
```

```
certmgr -add -7 -all -s CA CA.p7
```

```
certmgr -add -7 -all -s SPC SPC.p7
```

2. Transfer those three files to your home directory on the target UNIX system.
3. Become the superuser (root, use the su command).

4. Create the following directories:

```
/usr/local/certificates
```

```
/usr/local/certificates/CA
```

```
/usr/local/certificates/ROOT
```

```
/usr/local/certificates/SPC
```

5. Copy the p7 files to the appropriate directories:

```
cp CA.p7 /usr/local/certificates/CA
```

```
cp Root.p7 /usr/local/certificates/ROOT
```

```
cp SPC.p7 /usr/local/certificates/SPC
```

6. Make sure the files and directories have read-only access.

```
chmod -R 0555 /usr/local/certificates
```

7. Exit the superuser shell.

```
exit
```

Note: If you are not an administrator, ask your system administrator to do this for you. If the system administrator is unwilling, you can instead create the directories in your home directory. For example, if your home directory is /home/todd, you'd create the following directories:

```
/home/todd/certificates
```

```
/home/todd/certificates/CA
```

```
/home/todd/certificates/ROOT
```

```
/home/todd/certificates/SPC
```

After you have created those directories, you need to set

some environment variables. Users of sh based shells (sh, ksh, bash, zsh, etc.) would run the following commands:

```
ROOT_CERTIFICATES=/home/todd/certificates/ROOT  
export ROOT_CERTIFICATES  
CA_CERTIFICATES=/home/todd/certificates/CA  
export CA_CERTIFICATES  
SPC_CERTIFICATES=/home/todd/certificates/SPC  
export SPC_CERTIFICATES
```

Users of csh or tcsh would run:

```
setenv ROOT_CERTIFICATES /home/todd/certificates/ROOT  
setenv CA_CERTIFICATES /home/todd/certificates/CA  
setenv SPC_CERTIFICATES /home/todd/certificates/SPC
```

You can put those commands in your login file (.login for sh users, .cshrc for csh users) to always have them available when you log in.

Strong Encryption

PKWARE has added the ability to use strong encryption when compressing files. This new feature adds a higher level of security for your data. If your version of PKZIP is licensed to use this feature, you will be able to choose from several strong encryption algorithms to encrypt your files. Strong **encryption** is available as part of the optional security module. If your copy of PKZIP is not licensed to use this feature, contact PKWARE at 1-414-354-8699, or visit www.pkware.com, for information on how to activate strong encryption.

With strong encryption you can encrypt your files using AES 128, 192, 256 bit algorithms. You can configure PKZIP to use any of these algorithms.

- AES – 128 bit
- AES – 192 bit
- AES – 256 bit
- 3DES – 168 bit
- 3DES – 112 bit
- DES
- RC4 – 128 bit
- RC4 – 64 bit
- RC4 – 40 bit
- RC2 – 128 bit
- RC2 – 64 bit
- RC2 – 40 bit

To use strong **encryption**, specify cryptalgorithm or recipient options when you create your .ZIP files. Refer to online help.

IMPORTANT: PKWARE has introduced support for the use of strong encryption with .ZIP files. If you plan to share the .ZIP files you create using this strong encryption feature with others, you should first make sure the recipients have a compatible version of PKZIP 6.0 so they can decrypt your files.

The strong encryption feature lets you choose to use either a traditional password or an X.509 digital certificate for encryption. If you plan to use a digital certificate, you must first obtain and install a valid digital certificate onto your system. Visit PKWARE's website for information on where you can get a Digital Certificate.

File Capacities

The original .ZIP file format has faithfully met the needs of computer users since it was introduced by PKWARE in 1989. As computer technology has advanced over time, storage capacities have increased dramatically. These increases make the numbers and sizes of files that seemed unimaginable 10 years ago a reality today. In order to extend the utility of the .ZIP file format to meet these changing system needs, PKWARE has extended the .ZIP file format to support more than 65,535 files per archive and archive sizes greater than 4 Gigabytes (GB).

The specification for the .ZIP file format has been publicly available and distributed by PKWARE in a file called APPNOTE.TXT. This file documents the internal data structures and layout that define a .ZIP archive. The extensions introduced by PKWARE fully support all the features of your existing archives and newer versions of PKZIP that support these new extensions will continue to read all of your current archives.

Prior to the 4.5 release of PKZIP Suite, versions of PKZIP were limited to storing no more than 65,535 files in a .ZIP archive. Earlier versions of the original PKZIP for DOS could (and still can) store no more than 16,383 files per .ZIP archive.

Another limitation that existed prior to the 4.5 version of PKZIP Suite was that a single .ZIP archive could not be larger than 4 GB (4,294,967,295 bytes). Earlier versions of PKZIP for DOS cannot process .ZIP archives larger than 2 GB (2,147,483,647 bytes).

The extended .ZIP file format specification available with PKZIP Suite 4.5 and later supports creating .ZIP archives containing

over 4 billion files and having sizes larger than 9 quintillion bytes. These are only theoretical limits and most computer systems in common use today do not have enough storage capacity or available memory to create and store .ZIP archives approaching these limits.

The practical limits imposed by a typical computer in use today and configured with 128MB to 256MB of memory will support compressing up to approximately 262,144 files. Compressing this number of files can take a long time. Since it is not practical to reach the theoretical limits supported by the extended .ZIP file format, PKZIP does not currently compress or extract more than 2,147,483,647 files.

The actual limits on the numbers of files and sizes of archives can vary depending on the operating system you are using. The tables below list the capacities available in PKZIP for different operating systems.

Windows NT/2000/XP, Unix, Linux	Current Version
Archive size	9 EB *
Number of files in archive	2 G **
Number of segments for spanned / split archives	4 G
Size of segments for spanned and split archives	4 GB
PKSFX size	2 GB
PKSFX Patch size	2 GB

Windows 95/98/Me	Current Version
Archive size	4 GB *
Number of files in archive	2 G **
Number of segments for spanned / split archives	4 G
Size of segments for spanned and split archives	4 GB
PKSFX size	2 GB
PKSFX Patch size	2 GB

* Actual archive size depends on available storage space – the theoretical limit is 9 EB.

** Actual number of files may vary depending on available memory and resources. The theoretical limit is actually 4 G but the practical limit for most current system configurations is about 262,144 files.

Your available system resources and system settings limit the performance you can expect from PKZIP when processing large numbers of files or large archives. If you are compressing large numbers of files on a computer with insufficient memory, you can expect slow, or possibly incomplete processing if your available memory is depleted.

When compressing large files, it is a good idea to make sure you have your temporary folder set to a location on a drive with sufficient disk space available to process large files.

PKSFX self-extracting file sizes are limited to the size of an executable program file supported by the underlying operating system. This currently is 2 GB for most operating systems. This means you should not create a PKSFX file that is larger than 2 GB. PKZIP will be able to create PKSFX files larger than this limit. They will not, however, run on most operating systems. You can create and run split PKSFX files that are larger than 2 GB as long as each split segment is not larger than 2 GB.

Support for more than 65,535 files in an archive and archive sizes greater than 4 GB are features not available in versions of PKZIP released prior to PKZIP Suite 4.50. If you create archives that take advantage of these new features, you should be aware that older versions of PKZIP will not be able to recognize all of the files in the new archives you create that exceed the capacities of the older programs. If you plan to send a large archive to a friend or associate, they will need a compatible version of PKZIP in order to extract the contents of the file(s) you send.

Specific PKWARE products that do support these new features are:

- PKZIP V6.0 for Unix Command Line
- PKZIP V6.0 for the X Window System
- PKZIP V6.0 for Windows
- PKZIP V6.0 Command Line
- PKZIP Explorer V6.0

Versions of PKZIP prior to those noted above will not recognize these new features and will be unable to view or extract any files in your archives that are dependent on these features. Also, any .ZIP compatible programs you may be using from other companies will not be able to access all of the contents of your large archives. They may report that an archive is too large, or they may incorrectly report that the archive has errors. To ensure access to data in your large archives, always use genuine PKZIP.

Note: 4 GB is equal to 4,294,967,295 bytes.

9 EB is equal to 9,223,372,036,854,775,807 bytes.

Encoded and Archive File Support

PKZIP allows you to extract files from encoded or Tar and GZIP archived data. PKZIP decodes and extracts files created with the following encoding and archive schemes:

Encoded File Type	Common File Extensions
UUEncode	*.UUE
XXEncode	*.XXE
BinHex	*.HQX
MIME	*.MIM *.MME
Tar	*.TAR
GZIP	*.GZ *.TGZ
BZIP2	*.BZ *.bz2

In addition to decoding and processing the above file types, PKZIP allows you to encode .ZIP files in the UUEncode format.

To do so, select **Actions | UUEncode** from the menu.