Getting Started with PKZIP/SecureZIP for UNIX

On this page, you'll get *pkzipc* set up on your UNIX/Linux computer. You'll also learn more about the different editions of PKZIP and SecureZIP CLI, and confirming your purchase through license activation.

PKZIP Command Line and SecureZIP Command Line each come in both a Standard edition and an Enterprise edition. This table and the following sections describe the additional features included with SecureZIP Command Line that are not in PKZIP Command Line. They also describe the features added by the respective Enterprise editions of PKZIP Command Line and SecureZIP Command Line.

Feature	*PKZIP Standard (UNIX Only)*	*PKZIP Enterprise*	*SecureZIP Standard*	*SecureZIP Enterprise*
Large file size support	X	X	X	X
Very large archive support	X	X	X	X
Self-extracting files for end-users and other platforms		X	X	X
Decryption of PKI public-key encrypted ZIP archives	X	X	X	X
Attaching digital signatures to archives			X	X
Strong passphrase-based AES and 3DES data file protection			X	X
Strong encryption using a digital certificate instead of a passphrase			X	X
Strong, certificate-based file name encryption			X	X
Creating OpenPGP (RFC 4880) encrypted files			X	X
Opening OpenPGP files	X	X	X	X
Add digital timestamp from secure Time Stamp Authority			X	X
Error reporting for both attended and unattended operations	X	X	X	X
Email (SMTP) integration		X	X	X
FTP integration		X	X	X
Application Integration	X	X	X	X
Contingency Keys				X
LDAP Directory Integration				X

Product Varieties

SecureZIP Command Line Standard Edition on UNIX/Linux

On UNIX and Linux, SecureZIP Command Line Standard Edition adds the following features to the set provided by PKZIP Standard Edition:

- **Email and FTP integration**: Options to create and transfer archives by email or FTP directly from the command line. See "Sending an Archive by SecureFTP, FTP and Email."
- **PKSFX**: The ability to create self-extracting ZIP files for use in either the native command line or graphical Windows environment. See "Working with Self-Extracting (PKSFX) Archives."
- **Strong passphrase-based encryption**: Strong encryption—the kind of encryption used by banks and the federal government—is much more secure than the weaker, traditional ZIP encryption provided by PKZIP. See "Encrypting Files with a Passphrase."
- **Strong encryption using a digital certificate instead of a passphrase**: This kind of encryption is both more convenient and more secure than passphrase-based encryption, and it enables you to encrypt files just for the people you want to see them. See "Encrypting Files with a Recipient List."
- **Strong file name encryption**: With this feature, you can encrypt even the names of files in an archive so that only the intended recipients of the archive can read them. See "Encrypting File Names."
- **Digital signatures**: When you attach a digital signature, recipients of your files can be sure that the files are unchanged and really come from you. See "Attaching Digital Signatures."

PKZIP and SecureZIP Enterprise Editions

The Enterprise editions of SecureZIP and PKZIP command Line each add an additional module of functionality to the respective products.

PKZIP Enterprise Edition

PKZIP CLI Enterprise Edition includes the *Enhanced Data Processing* module. This module adds these features to PKZIP (all are included in SecureZIP):

- Email and FTP integration: Options to create and transfer archives by email or FTP directly from the command line. See "Sending an Archive by SecureFTP, FTP and Email."

- PKSFX: The ability to create self-extracting ZIP files for use in either the native command line or graphical Windows environment. See "Working with Self-Extracting (PKSFX) Archives."

SecureZIP Enterprise Edition

SecureZIP Enterprise Edition includes the *Directory Integration* module. This module enables SecureZIP to access digital certificates stored on directory servers anywhere in the enterprise. Being able to access certificates on directory servers makes it much more convenient to do strong certificate-based encryption, as you can encrypt for a set of recipients without needing to have the certificate for each recipient on your own machine. See "Accessing Recipients in an LDAP Directory."

SecureZIP Enterprise Edition also includes the *Contingency Keys* module. Contingency keys are digital certificate-based keys that an administrator can have automatically included in the recipient list whenever PKZIP does strong encryption. See "Contingency Keys" for more information.

SecureZIP Enterprise Edition provides additional functionality regarding OpenPGP keys and X.509 certificates. These include:

- Ability to generate OpenPGP keys
- Convert X.509 certificates to OpenPGP keys
- Convert OpenPGP keys to X.509 certificates
- Signing OpenPGP keys

If you are transitioning from the McAfee eBusiness Server (EBS), you can use SecureZIP Enterprise Edition in OpenPGP Mode to run many of your existing EBS scripts with minimal editing. The commands include ***decrypt***, ***encrypt***, and ***sign***. These commands and options are described on this page.

Find more information about Enterprise Edition features, including installation and a command reference, in the *Getting Started with PKWARE Key Maker* guide.

Learning More and Getting Help

This manual is not the only way to learn about PKZIP and SecureZIP. You can find additional information inside the program itself, and on the World Wide Web.

Using Help

PKZIP provides a help system for the PKZIP commands and options. The help system describes syntax and shows sample command lines.
Access the help system directly from the command line:

- At the command prompt, type the following and press ENTER:

  pkzipc -help

  A screen with PKZIP version and usage information appears. You can get help for any PKZIP command or option from here.

- To bypass the command/option menu and go directly to a help file for a command or option, type the ***help*** command followed by an equal sign (**=**) and the command or option for which you want information.

  For example, to access online help for the ***add*** command, type the following at the command prompt and press ENTER:

  pkzipc -help=add

  The help information for the ***add*** command appears.

Getting Version Information

version

To list the version of PKZIP that you are using, use the ***version*** command:

pkzipc -version

This command line outputs two lines like the following after the usual header information:

```
Program File Version (pkzipc): 12.50.1087
Product Version: 12.50.0005
```

The first line lists major, minor, and step version numbers of the *program*:

```
Program File Version (pkzipc): <major>.<minor>.<step>
```

The second line lists the major and minor version numbers and the build number of the *product*.

```
Product Version: <major>.<minor>.<build>
```

Major and minor version numbers of the program are always the same as those for the product.

In addition to producing this display output, the ***version*** command returns a version number as a value to the shell. The version number returns as a positive integer value less than 256. This value is only returned to the shell and is not displayed in normal output. It can be used to verify PKZIP version numbers in a .BAT file or shell script.

Sub-options of the ***version*** command (described in the following table) determine which version number is returned. The major version number is returned by default.

Sub-Option	*PKZIP Returns*	*For example*
major	The major release number. For example, if the version number is 12.10.1054, the value returned is 12. This is the default return.	***pkzipc -version*** ***pkzipc -version=major***
minor	The minor number of the release. For example, if the version number is 12.10.1054, the value returned is 10.	***pkzipc -version=minor***
step	The step or patch value (minus 1000 if 1000). For example, if the program version is 12.10.1054, the value returned is 54.	***pkzipc -version=step***
product	The build number of the product. For example, if the product version is 12.10.0003, the value returned is 3.	***pkzipc -version=product***

Technical Support

For support, visit our Web site at: https://support.pkware.com/

Working With Your License

Entering License Keys

Note: To use SecureZIP Partner, as a participant in PKWARE PartnerLink, *you do not need to enter a license key*. You can ignore this section and related sections on getting license information and sharing a license, later in this chapter.

You must enter your license key to activate the product and for any add-on modules after you complete the installation.

You must run PKZIP as root to use the ***enterlicensekey*** command. If you try to run the command as an ordinary user instead of as the super user, you get an error.

Running the enterlicensekey command creates a license.ini file (if it does not exist already) in the PKZIP installation directory where the `pkzipc` executable is located. The license file must be in this directory for PKZIP to find it. The default location of this directory is:

- ***/opt/pkware/pkzip/bin/*** on Solaris and HP-UX
- ***/usr/pkware/pkzip/bin/*** on AIX and Linux.

Make the directory and its files readable for all users and writable for none.

You can use the ***enterlicensekey*** command to enter license keys on Windows as well. You may want to do this if you need to enter the license key for an add-on module that you purchase sometime after you purchased the base product.

To enter a license key:

1. Become the super user to run the program as root.
2. At the command prompt, type the following and press ENTER:
   pkzipc -enterlicensekey
   PKZIP prompts you for a product license key.
3. Enter a product license key and press ENTER.

Repeat these steps for each license key you have. For example, if you have a license key for an add-on module, repeat the steps above to enter the license key for that module after you enter the license key for the base product.

Getting License Information

To display the PKZIP license information on your screen, type ***pkzipc -license*** at the command prompt and press ENTER

Notes for UNIX Users

Using Wildcards with PKZIP on UNIX

If your shell is set up to automatically expand wildcards, you should put file specifications that use wildcards—for example, `*.htm`— in quotation marks—like this: `"*.htm"`—on the command line to prevent the shell from expanding them.

Allowing the shell to expand wildcard file specifications into an explicit list of files can cause the PKZIP ***recurse*** and ***directories*** options not to work properly. Placing a wildcard pattern in quotes instructs the shell to pass the pattern as an argument to PKZIP, which then expands it.

PKZIP can interpret and expand the following wildcard patterns:

Pattern	*Example*
`*`	`*`
<pattern>	`*.txt, *f.txt`
<pattern>*	`h*, file.f*`
<pattern> <pattern>*	`a*.txt`
<pattern>	`"*.*", *ab*`

Running the Program as Root

Setting the ***set-uid*** bit on the pkzipc binary causes PKZIP to run as root. It also causes PKZIP to run any program that it may launch—such as the ftp client (***ftp*** option)—as root.

Use considerable caution in setting the ***set-uid*** bit to run PKZIP as root. It is very easy for a program running as root to overwrite system files, and setting the ***set uid*** bit on any program raises security concerns.

Configure PKZIP to run this way only in keeping with organizational security policies and on the instructions of a system administrator.

Information for PartnerLink™ Sponsors and Partners

PKWARE PartnerLink enables a *sponsor* organization that has SecureZIP to distribute to *partner* organizations the SecureZIP Partner application. That is, an organization that licenses SecureZIP can exchange strongly encrypted archives with selected Partner organizations, offering both sponsor and partner organizations secure business-to-business communication. SecureZIP Partner is a special version of SecureZIP. It provides most of the commands and options of SecureZIP but works only with archives created by (or for) a sponsor. Archives created using SecureZIP Partner are automatically strongly encrypted for sponsor recipients.

This section applies only to participants in the PKWARE PartnerLink program, including users of SecureZIP Partner. Other readers may skip this section.

Note: SecureZIP Partner was called *SecureZIP Reader/SecureLink* prior to release 8.5 of SecureZIP Server.

To use SecureZIP Partner, *you do not need to enter a license key*. Use of the software is controlled by the Sponsor Distribution Packages you install. Users of SecureZIP Partner can ignore the section "Entering License Keys" and related sections on getting license information and sharing a license.

If You Are a Sponsor: Sign the Central Directory

A sponsor organization uses SecureZIP as usual to work with archives for, or from, a partner. There is just one special requirement when creating an archive for a partner: you must sign the central directory of the archive using a certificate included in the Sponsor Distribution Package (SDP). Otherwise a partner cannot extract the archive.

To sign an archive, use the ***certificate*** option. (See "Attaching Digital Signatures.") You may optionally sign files in addition to signing the archive itself. Use the ***sign*** option to specify what to sign: the central directory, the archive's files, or both.

For example, the following command line adds files to archive `test.zip`. The command line signs using the *John Q. Public* certificate and attaches the signature to the central directory only, not to the archive's files.

```
pkzipc -add -certificate="John Q. Public" -sign=cd test.zip *.*
```

Contact PKWARE for information about participating in the PartnerLink program or assembling a Sponsor Distribution Package for partners.

If You Are a Partner

A PartnerLink partner uses the SecureZIP Partner application to work with archives. The SecureZIP users manual you are now reading also serves as a user guide for SecureZIP Partner.

See the *PartnerLink Partner Setup Guide: Windows/UNIX/Linux* for information on installing SecureZIP Partner and on setting up as a partner to work with sponsor archives.

About SecureZIP Partner

SecureZIP Partner does basically two kinds of operations:

- **Extracts files from sponsor archives:** SecureZIP Partner uses SecureZIP commands and options to extract files from a ZIP archive received from a sponsor. These commands and options include those to decrypt and decompress files and to authenticate digital signatures. SecureZIP Partner can only extract archives digitally signed by a PartnerLink sponsor.
- **Creates archives for sponsors:** SecureZIP Partner uses SecureZIP commands and options to add files to a ZIP archive, including commands and options to compress, encrypt, and digitally sign files.

SecureZIP Partner can create and update archives only for a designated sponsor. Archives are automatically encrypted for all sponsor recipients whose certificates are included in the sponsor's SDP. Only those sponsor recipients can decrypt and read the files in an archive created by SecureZIP Partner. SecureZIP Partner does not use passphrase-based encryption.

Note: Because SecureZIP Partner automatically encrypts for sponsor recipients—and only for sponsor recipients—when adding files to an archive, partners cannot decrypt archives that they use SecureZIP Partner to create. So partners must be careful not to delete files they want to keep after placing them in an archive. A copy of a file in an archive will be inaccessible to the creator of the archive.

To Run SecureZIP Partner

The command to run SecureZIP Partner is ***pkzipr*** ; the command to run SecureZIP is ***pkzipc*** . So, for example, where the manual says to use a command like the following to extract all files from archive myfiles.zip:

```
pkzipc -extract myfiles.zip
```

you would instead use a command line like one of those below to do the same thing with SecureZIP Partner:

```
pkzipr -extract myfiles.zip
pkzipr -extract -sponsor="Example Corp" myfiles.zip
```

Designating a Sponsor

SecureZIP Partner only operates on archives from, or for, a sponsor. A special ***sponsor*** option is provided just for SecureZIP Partner to designate a sponsor.

sponsor

The ***sponsor*** option is only required when creating or updating an archive with the ***add*** command. The option can be explicitly included on the command line, or you can configure SecureZIP to designate a sponsor by default (see "Changing Defaults for Commands and Options - UNIX").

You do not need the ***sponsor*** option when extracting an archive with the ***extract*** command. If the option is not used when extracting, the signature on the archive is checked against all sponsors defined on the system.

Use the ***sponsor*** option when extracting if you want to ensure that only an archive from the specified sponsor is extracted. For example, you may have a script to process archives from a particular sponsor. Use the ***sponsor*** option with command lines in the script to ensure that the script does not inadvertently process an archive from some other sponsor.

You can use the ***sponsor*** option multiple times on the same command line when extracting but only once per command line when adding files to an archive.

The ***sponsor*** option accepts either a sponsor's common name or sponsor ID to identify a sponsor. To find out this information about a sponsor, use the PKSponsor ***list*** command or the SecureZIP Partner ***listSponsors*** command, to list sponsors. (PKSponsor is a tool included with SecureZIP Partner for setting up as a partner. See the *PartnerLink Partner Setup Guide: Windows/UNIX/Linux*.)

For example, the following command line adds files to a ZIP archive for sponsor Example Corp. It references Example Corp by common name:

```
pkzipr -add myfiles.zip -sponsor="Example Corp" *.doc
```

The similar example below uses the sponsor ID to reference a sponsor:

```
pkzipr -add myfiles.zip -sponsor=15 *.doc
```

If you have received an archive called `myfiles.zip`, and you are uncertain which of your sponsors it came from, use the ***sponsor*** option twice to extract the archive's files from either sponsor:

```
pkzipr -extract -sponsor="Example Corp" -sponsor=20 myfiles.zip
```

Listing Available Sponsors

SecureZIP Partner provides a ***listSponsors*** command to list sponsors, like the PKSponsor ***list*** command.

listSponsors

The following command line returns a list of sponsors on the system:

```
pkzipr -listsponsors
```

Output from ***listSponsors*** looks like this:

```
----- Sponsor #1 -----

        Sponsor: PKWARE, Inc.
        Sponsor ID: 0
        Type: Read/Write
        Description: <Sponsor 1's comment, if any>
----------------------------------

----- Sponsor #2 -----

        Sponsor: ABC Corp
        Sponsor ID: 1
        Type: Read/Write
        Description:
------------------------
2 sponsor(s) installed
```

The table below explains the fields.

Field	Description
Sponsor	Common name of a sponsor
Sponsor ID	ID of a sponsor
Type	Functionality profile. *Read/Write* indicates that functionality is supported both for extracting sponsor archives and for creating archives for sponsors.
Description	Optional comment of sponsor

Commands and Options Available with SecureZIP Partner

SecureZIP Partner enables you to use nearly all SecureZIP commands and options. Only a few cannot be used, generally because they cannot be constrained to work only with archives created by or for a sponsor.

The SecureZIP commands and options that you cannot use are listed in the following table.

Commands and options *not* available in SecureZIP Partner

ArchiveType	MailTo*	SfxDirectories
Encode*	NameSfx	SfxLogfile
EnterLicenseKey	NoFix	SfxOverwrite
Fix	Recipient	SfxUIType
FTP*	RunAfter	VerifySigner
LDAP	Sfx	
ListSfxTypes	SfxDestination	

Notes:

- Items flagged with an asterisk in the table above have both a command form and an option form. The command form is not available in SecureZIP Partner. See "Understanding Commands and Options" in Chapter 1 for more information.
- The ***view*** command does not work on archives that you create for a sponsor using encrypted file names (see the ***cd*** option).