

Sending an Archive by SecureFTP, FTP and Email

This chapter describes the command line options to transfer a new or existing archive to other people by FTP or email. You need PKZIP Enterprise or SecureZIP for this functionality.

Transferring an Archive with FTP

ftp

If your machine has a standard FTP (File Transfer Protocol) program to transfer files over the Internet, you can include an instruction to PKZIP to use the program to send an archive after creating it. For example, the following command lines each create an archive `mydocs.zip` and transfer it to the address specified in the `ftp` sub-option. The second example explicitly specifies an FTP user name, passphrase, and account:

```
pkzipc -add -ftp=wash/home/thomas mydocs.zip *.doc
```

```
pkzipc -add -ftp=jefferson:monticello:vip@wash/home/thomas mydocs.zip *.doc
```

The `ftp` command/option can be used with the `add` command, as in the command lines above, or by itself. When used as a command by itself, `ftp` simply transfers the specified file. For example, the following command line transfers existing file `mydocs.zip`:

```
pkzipc -ftp=jefferson:monticello@wash/home/jefferson mydocs.zip
```

Use `ftp` with the `delete` command to transfer an archive after deleting some files in it:

```
pkzipc -delete -ftp=wash/home/jefferson mydocs.zip *.txt
```

You can configure `ftp` to use a default address, but you must still include the option on the command line to actually perform an FTP transfer.

```
pkzipc -add -ftp mydocs.zip mydocs.zip *.doc
```

The `ftp` address sub-option has the following syntax (optional fields are bracketed).

- To specify a full path on the server:

```
-ftp=[username[:passphrase[:account]]@]server//fullpath
```

- To specify a relative path on the server, that is, a path relative to the directory that the server chooses for your login:

```
-ftp=[username[:passphrase[:account]]@]server/relpath
```

where:

- `username` (optional) is the user account with which to log in if the FTP server requires a login. If a username is not supplied, PKZIP tries to log in as the user `ftp`.
- `passphrase` (optional) is the passphrase associated with the user account. If no passphrase is given, PKZIP tries an empty passphrase. A colon is not allowed in the passphrase as this character is used to separate username, passphrase, and account values.
- `account` (optional) is for use only with FTP servers that require additional authentication. Do not specify the account for servers that do not require it.
- `server` is the FTP server name
- `path` (relative path or full path; optional) is the path to the destination of the transferred file on the server. If you omit a path, PKZIP transfers the archive to the default folder on the FTP server.

Include the `movearchive` option to delete from your hard disk an archive that you no longer want after transferring it:

```
pkzipc -add -movearchive -ftp=wash/home/jefferson mydocs.zip *.doc
```

If for some reason an archive is not transferred to the FTP server, `movearchive` does not delete it.

Note: The `ftp` option can only send ZIP archives that are physical files. It cannot send ZIP archives from STDIN, STDOUT, or special files (named pipes, sockets).

Securely Transferring an Archive with SFTP

sftp

UNIX users may also use the SSH File Transfer Protocol (`sftp`) to send ZIP archives through a secure channel. It works exactly as ftp does, as shown in this example:

```
pkzipc -add -sftp=jefferson:monticello:vip@wash/home/thomas mydocs.zip *.doc
```

Sending an Archive by Email

mailTo, *mailFrom*, *mailServer*, *mailBCC*, *mailBody*, *mailCC*, *mailOptions*, *mailReplyTo*, *mailSubject*

You can send a new or existing archive as an email attachment directly from the PKZIP command line. To do so, use the *mailTo* option to specify recipients of the message, *mailFrom* to give your own address, and *mailServer* to list the SMTP server to use to send the message. Other options are available for such other common email-related fields as CC (for recipients to be sent a copy) and BCC (for recipients to be sent a blind copy).

For example, the following command line adds files to archive data.zip and emails the archive to John Public as an attachment:

```
pkzipc -add -mailTo=john.public@abc.com -mailFrom=me@myplace.com -mailServer=smtplib.myplace.net -mailSubject="Latest sales" data.zip *.doc
```

In the following example, *mailTo* is used as a standalone command, without *add*, to send an existing archive:

```
pkzipc -mailTo=john.public@abc.com -mailFrom=me@myplace.com -mailServer=smtplib.myplace.net -mailSubject="Latest sales" data.zip
```

You can include the *movearchive* option to delete from your hard disk an archive that you no longer want after emailing it.

Note: The *mailTo* command/option can only mail ZIP archives that are physical files. It cannot mail ZIP archives from STDIN, STDOUT, or special files (named pipes, sockets).

Configuring Required Options

To email an archive, each of the three options *mailTo*, *mailFrom*, and *mailServer* must be specified.

To avoid having to specify these three options on the command line, you can use the *configuration* command to configure values for *mailFrom* and *mailServer* for use by default. Then you need only specify *mailTo* on the command line. All the *mail...* options are configurable. (To tell PKZIP to mail an archive, you must include *mailTo* on the command line even if a value for the option is configured.)

Specifying a Mail Server

The *mailServer* option specifies the SMTP server to use. The server specified for *mailServer* must be available without a proxy server and must allow email to be forwarded from the machine on which you run PKZIP.

Set the name or IP address of the server into *mailServer* as a sub-option. You can either do this on the command line, as in the preceding examples, or you can configure *mailServer* to use a specified server by default. For example:

```
pkzipc -config -mailserver=mail.abc.com
```

If necessary, you can specify a user name and/or passphrase. This tells PKZIP to try plain-text or login authentication to connect to the server. Prefix the passphrase with a colon, and use an *at* sign (@) to separate user/passphrase information from the server address like this: user:passphrase@server. For example:

```
pkzipc -config -mailserver=john.mypassword@mail.abc.com
```

```
pkzipc -config -mailserver=:mypassword@mail.abc.com
```

Note the colon before the passphrase.

The following command line creates and sends data.zip with the message text specified in *mailBody*. Set off the message text in quotes:

```
pkzipc -add -mailTo=john.public@abc.com -mailSubject="Latest sales" -mailBody="Here are the sales figures I promised." data.zip *.doc
```

Sending to Multiple Recipients

To send an archive to multiple email recipients, use *mailTo* multiple times or use it to specify a file that lists recipients. The following command line uses *mailTo* multiple times to send to multiple recipients. Each receives a message listing all other recipients who appear in the TO list:

```
pkzipc -add -mailTo=john.public@abc.com -mailTo=jane.doe@abc.com -mailSubject="Latest sales" -mailBody="Here are the sales figures I promised" data.zip *.doc
```

Sending to a List of Recipients

The *mailTo* option can take the name of a list file as a sub-option. In the file, list addresses of recipients one to a line. On the command line, prefix the file name with the *listchar* character (@ by default). The message is sent to every address in the file:

```
pkzipc -add -mailto=@addresses.txt -mailserver=mail01 -mailfrom=sam.adams@wash.com files.zip *.doc
```

Sending Encrypted Attachments

Note: The *recipient* option is available only in SecureZIP.

Use **mailTo** with its **recipient** sub-option to send an archive to the same recipients for whom you encrypt it. For example:

```
pkzipc -add -recipient=tom.jefferson@wash.com -recipient=sam.adams@wash.com -mailTo=recipient -mailserver=mail01 -mailfrom=sam.adams@wash.com files.zip *.doc
```

The command line above uses the **recipient** option to encrypt the archive for specified recipients. It uses **mailTo** with the **recipient** sub-option to send the archive to those same recipients.

Note: For the **mailTo recipient** sub-option to work, the recipients' certificates used to encrypt must contain email addresses. PKZIP alerts you with a warning message for any recipient for whom PKZIP cannot find an email address.

The **recipient** sub-option of **mailTo** can be used only when **mailTo** is used as an option with another command such as **add**; the **mailTo recipient** sub-option cannot be used when **mailTo** is used as a standalone command.

If you use the **recipient** option (not the **mailTo recipient** sub-option) to specify a file that lists the names of certificate holders, you do not need to list recipients on the command line. In this case, using **mailTo** with the **recipient** sub-option encrypts for, and sends to, all the certificate holders in the list, using the email addresses associated with their certificates.

```
pkzipc -add -recipient=@addresses.txt -mailto=recipient -mailserver=mail01 -mailfrom=sam.adams@wash.com files.zip *.doc
```

Specifying Text in a File

If the text of the subject or body of the message is more than a few words or contains quotes, you can put the text in a file and specify the file in the sub-option. For example:

```
pkzipc -add -mailTo=john.public@abc.com -mailTo=jane.doe@abc.com -mailSubject=@subject_text.txt -mailBody=@body_text.txt data.zip *.doc
```

Sending Copies

Use **mailCC** and **mailBCC**, respectively, to specify recipients to receive copies (CC) and blind copies (BCC) of messages. You can specify recipients' addresses directly, or you can specify a file containing a list of addresses.

Each recipient in the following command line receives a message showing all **mailTo** names in the TO list and the **mailCC** recipient in the CC list:

```
pkzipc -add -mailTo=john.public@abc.com -mailTo=jane.doe@abc.com -mailSubject="Latest sales" -mailBody="Here are the sales figures I promised" -mailCC=rich.smith@abc.com -mailBCC=bill.cody@abc.com data.zip *.doc
```

To send copies or blind copies to multiple recipients, either use **mailCC** or **mailBCC** multiple times, or list recipients in a file. Prefix the file name with the list character:

```
pkzipc -add -mailTo=john.public@abc.com -mailTo=jane.doe@abc.com -mailSubject="Latest sales" -mailBody="Here are the sales figures I promised" -mailCC=rich.smith@abc.com -mailCC=bill.cody@abc.com -mailBCC=@address_list.txt data.zip *.doc
```

Sending Split Archives

If you use the **span** option with **mailTo** to create and mail a split archive, PKZIP sends each segment of the split archive in a separate mail message. This is useful when your recipient's email server has size limits on file attachments on individual emails.

Hiding the TO List

If you do not want recipients to see names of other recipients in the TO list, use the **mailOptions** option with either the **each** or the **undisclosed** sub-option.

The **each** sub-option causes each **mailTo** recipient to receive a message showing only his own name in the TO list. All **mailTo** recipients see all names in the CC list. Any **mailCC** and **mailBCC** recipients receive a copy of each message to each **mailTo** recipient:

```
pkzipc -add -mailTo=john.public@abc.com -mailTo=jane.doe@abc.com -mailOptions=each -mailSubject="Latest sales" -mailBody="Here are the sales figures I promised" -mailCC=rich.smith@abc.com -mailBCC=bill.cody@abc.com data.zip *.doc
```

The **undisclosed** sub-option works just like the **each** sub-option except that the message that each recipient receives displays Undisclosed in the TO field instead of the recipient's name.

The **each** option causes PKZIP to generate a distinct mail message for each recipient, showing only that recipient's address in the TO field. The **undisclosed** sub-option requires PKZIP and the mail server to do less processing and so sends a bit faster.

Including Instructions on How to Unzip

The **instructions** sub-option of **mailOptions** causes PKZIP to include a small, additional attachment explaining how to unzip a ZIP file.

```
pkzipc -add -mailTo=john.public@abc.com -mailSubject="Plans" -mailOptions=instructions plans.zip *.doc
```

The *instructions* and *each* sub-options of *mailOptions* can be set together, separated by a comma:

```
... -mailOptions=each,instructions ...
```

Using a ReplyTo Address

With the *mailReplyTo* option, you can specify an alternate email address for recipients to use to reply to the message instead of the *mailFrom* address. For example:

```
pkzipc -add -mailTo=john.public@abc.com -mailFrom=jane.doe@xyz.com -mailSubject="Plans" -mailreplyTo=jane.doe@myplace.net  
plans.zip *.doc
```