Encryption

Encryption is only available in versions of ZipIt registered to users in the United States, due to the United States export law. You must register your copy of ZipIt to get encryption features; see the section entitled “Registration,” above, for details. With encryption, you can protect a file so that it will be impossible to unzip without the proper password.

Encrypting a file. First, add the file to a zip window, if the file is not already there, by choosing “Add…” from the Zip menu. Next, select the file that you want to encrypt. You can encrypt files that have already been saved in the zip archive if you wish. Choose “Set Password…” from the Zip menu. (This choice will only be available on versions of ZipIt that have encryption enabled.) Type in the password you want to use, then re-type your password in the second field. Click OK, and a lock icon will appear next to the file. That means that the entry will be encrypted as soon as you save the zip archive.

Unzipping and decrypting a file. If you try to extract an encrypted file, ZipIt will ask you for the password to use when decrypting it. Simply type in the correct password, and ZipIt will be able to unzip the file. ZipIt will remember the password until you close the zip archive, so if you need to unzip it again you do not need to re-enter the password. You may also tell ZipIt the password in advance by selecting the encrypted files and choosing “Set Password…” from the Zip menu. If more than one encrypted file in a zip archive is encrypted with the same password, simply select all the files that use that password, choose “Set Password…” and type the password. This prevents you from having to enter the same password repeatedly.

Decrypting a file without unzipping it. You can remove the password from a zip entry while keeping it in the zip archive. After doing this, and saving the zip archive, a password will no longer be needed to unzip the entry. Select the entry that you want to decrypt (it must have a lock icon next to it, indicating that it is encrypted) and choose “Remove Password…” from the Zip menu. Unless the files that you chose were not yet saved, you will be prompted for the password to the entries. Enter the password, and the lock icon will change to an open lock. This open lock icon signifies that ZipIt will decrypt the entry as soon as you save the zip archive. Note: ZipIt does not check to see if your password is correct when you enter it. If you type the wrong password, ZipIt will not be able to decrypt the file and, after the zip archive is saved, the entry will remain encrypted.

MacBinary

MacBinary a protocol that was originally designed to facilitate file transfers of Macintosh files. It is a method of storing all the information necessary to recreate an entire Macintosh file in one data stream, instead of the two forks plus extra file information that the Macintosh normally uses. ZipIt uses MacBinary to store Macintosh files inside a zip archive.

Zip archives created with ZipIt are always compatible with other zip implementations. However, if you are zipping a file intended for use by a computer other than the Macintosh, you should not use MacBinary when zipping that file. If you do, then, after unzipping the file on the other computer, the resulting file will have extra Macintosh data; specifically, 128 bytes of garbage will appear at the beginning of the file when it is unzipped on the other machine.

You can control whether or not ZipIt uses MacBinary on a specified file by simply clicking in the “MB” column in the zip window. If there is a hollow circle there, then MacBinary will not be used to zip the file. If there is a filled circle, MacBinary will be used. You can only change the MacBinary status of a file before it has been saved. You cannot convert a file to MacBinary or vice-versa after it has been zipped and stored in the archive.

File names. One consequence of MacBinary usage is that ZipIt keeps track of two separate filenames for each entry. The first filename is the name as it is written in the zip archive. For example, if you zipped a Macintosh application called “Cool World.sit,” ZipIt would store the name as `COOLWORL.SIT`. If another zip program opened this zip archive, it would think that the name of Cool World was `COOLWORL.SIT`. This is useful because computers running operating systems such as DOS could then unzip the file. ZipIt, however, will always know that the name of the file is Cool World.sit. So whenever you open the file with ZipIt, you will see that the name is Cool World.sit.

Note: You can turn off ZipIt’s “dosification” of filenames by deselecting the appropriate option in the Miscellaneous Preferences dialog box. In that case, ZipIt would store the name of the above file as “Cool_World.sit”. Spaces, slashes, and extended characters are converted to underscores in this case. This is useful if the destination machine is a Unix, and not an IBM.

You can see the MacBinary feature in action. Add any Macintosh file to a zip window and click on the MacBinary oval. When MacBinary is on, you will see the “real” Macintosh name. When MacBinary is off, you will see ZipIt’s best approximation of the Macintosh name in a format acceptable to the IBM (assuming that the filename conversion option is on).

You can control how ZipIt decides whether to use MacBinary. Choose “Miscellaneous Preferences” from the File menu. You will see a choice, “Use MacBinary: Always, When needed, Never.” If you choose “Always,” then any file you add will have the MacBinary indicator filled in. If you choose “Never,” then ZipIt will never fill in the MacBinary dot. If you choose “When needed,” ZipIt will only use MacBinary if the file has a resource fork. You can always change whether or not MacBinary will be used on an unsaved file by clicking on the dot in the MacBinary column, no matter what the settings in the preferences dialog are.

MacBinary and other zip programs. Suppose you store a file in an archive, using MacBinary. You then send the archive to a friend who has an IBM; he adds a file

(obviously not using MacBinary). He then sends the archive back to you. The original file that you zipped will be intact, with MacBinary information fully functional. Suppose instead that your friend unzipped the file that you sent him, then re-zipped it and sent you back the archive. In this case, ZipIt will no longer recognize the file as MacBinary. In order to get it back, you would have to unzip it, then choose “MacBinary…” from the Translate menu. Suppose that you sent your IBM friend some files inside a MacBinary folder. Your friend then adds more files to the archive. Because of the way ZipIt stores MacBinary information for folders, that information will be lost if the archive is re-saved, even if the folders are not touched. Specifically, the name of the folder that was originally MacBinary will revert to its DOS form. In summary: MacBinary files can survive if the original zip file is modified by another program; MacBinary folders cannot.

Force MacBinary. In version 1.3, you may force a file’s MacBinary status to be on or off. If there is a file already in an archive, and the file is in MacBinary format, but was not created with ZipIt, you may tell ZipIt to automatically convert that file when unzipping it. Hold down the option key while clicking in the MB button to activate this feature. The filename used when unzipping will always be the original MacBinary filename.

Text and Binary Files

Linefeeds

On the Macintosh, a carriage return character is normally used to indicate the end of a line (or, in word processors, the end of a paragraph). This is not the case on other computers. On IBMs and compatibles, a carriage return is always followed by a linefeed, and on Unix systems, a sole linefeed is used. ZipIt will convert text files using either of these two formats to Macintosh format automatically, during decompression. ZipIt will usually be able to detect when this must be done. However, if you would like to manually toggle this feature, simply select the files and choose “Add/Strip Linefeeds” from the Zip menu.

ZipIt will also add linefeeds to text files destined for non-Macintosh systems. You must tell ZipIt to add linefeeds before the file is zipped; ZipIt cannot add linefeeds to a file once it is already saved. You can turn this option on by clicking in the linefeed square (the square in the “LF” column) in the zip window next to the file to which you want to add linefeeds, or by selecting that file and choosing “Add/Strip Linefeeds” from the Zip menu. If the square is filled in, linefeeds will be added when the file is saved. By default, linefeeds will be added to all text files if the Preference for adding and stripping linefeeds is set.

Important: Suppose you create an archive. Your IBM friend successfully unzips it, but cannot use the files inside. The first thing you should try is to toggle the setting of the “Add/Strip Linefeeds” menu command (or the LF button). Similarly, if you have a zip archive that contains files that you can successfully unzip, but cannot use, change this setting.

Distinguishing Between Text and Binary Files

In order to determine whether or not to strip linefeeds from a file, ZipIt normally examines the information left by the program that originally created the archive. If that program marked a specific file as text, then ZipIt will strip linefeeds; otherwise, it will not. However, such programs often make mistakes. If you find that you receive many files with a certain extension, say, `FILE.DOC`, and that those files are invariably text files, even though they are not marked as such, then you may override the demarkation and allow ZipIt to recognize these files as text. (You may also force ZipIt to recognize these files as binary.) Add the files to the Extension mapping preference dialog box, and either set or leave unset the “Text File” checkbox. See the “Extension mapping” subsection of the “Preferences” chapter, below.

Folders

ZipIt version 1.2 introduced a new feature: support for folders. On the IBM, folders as such normally do not exist in zip archives. Directories (as they are called in the DOS world) are implied by pathnames. For instance, a file might be called `FOLDER1/FOLDER2/FILE.TXT`. This would mean that a file, called `FILE.TXT`, exists inside a directory called `FOLDER2`, that in turn is inside a directory called `FOLDER1`. ZipIt interprets this information as it reads the zip archive, and displays folders as such. To continue our example, if there were only one entry in the zip archive, when you opened the zip archive you would see a single folder, `FOLDER1`. Double-clicking on that folder would open it, to show you `FOLDER2`. Double-clicking on that folder would open it, to show you the file. You can extract an entire folder by selecting it and choosing “Extract…” from the Zip menu.

MacBinary has a special meaning when applied to folders. ZipIt needs to store extra information about folders, such as their full names, and their creation dates. To do this, ZipIt places extra information at the end of the zip archive. This information is only stored if MacBinary is turned on for a folder. Although it is still called MacBinary, this information is not in standard MacBinary format, and cannot be read by any program other than ZipIt. MacBinary can be turned on and off for a folder even after the zip archive is saved.

Segments

The major new feature in ZipIt 1.3 is its support for multi-segment zip archives. This support is implemented in the following way. When opening existing multi-segment archives, for instance, created with PKZip 2.04g, you must open the final segment. ZipIt will then read the entire directory, and you will be abel to unzip any of the files. ZipIt will prompt you when it needs a different segment. Due to the structure of the zip archive, there is no way for ZipIt to know if you give it the wrong segment; therefore, if this happens, the most likely result is that the zipping process will end and ZipIt will report

that the entry is corrupted.

To create a multi-segment archive, simply save the archive to a floppy disk. When ZipIt runs out of room on that disk, it will ask you where you want to save the next segment. You can segment an existing archive by choosing “Save As…” from the File menu and saving it to a floppy disk.

You can also set the segment size using the appropriate option under the Zip menu.

PKUnzip and perhaps other programs as well require that the filenames for each of the segments be the same. In addition, the name of each disk that contains a segmented archive must be PKBACK# XXX, where XXX is the number of the segment, starting at 001.

Note: You may have problems saving to a PC disk mounted with PC Exchange. If you do have problems, then you will need to save segments to your hard disk (using the Set Segment Size option under the Zip menu), and then manually copy the segments to PC floppy disks. If you do have problems, please let me know.

Drag and Drop

In version 1.3.3 and later, you can now drag and drop files into and out of archives, as well as between archives and within an archive, assuming you have Macintosh Drag and Drop (which comes with System 7.5). As a shortcut to unzip a file from an archive, you can simply drag it from the archive to the folder on the desktop where you want it to be unzipped. Similarly, to add a file from the desktop to an archive, just drag it into the archive. (You will still need to save the archive in order for the file to be compressed.)

You can also drag compressed files between archives. If you do this, the file is first unzipped, and then it is added to the new archive. The new archive will need to be saved (and the file re-compressed) in order for your changes to take effect.

You can drag files within a single archive, in order to move them into and out of folders. To move a file into a folder, simply drag it to that folder. To move a file out of a folder and into the enclosing folder, drag the file up to the popup menu at the top of the archive window.