Scripting ZipIt

Starting with version 1.3.7, ZipIt is fully scriptable. This means that you can do anything via AppleScript that you can manually. The scripting capability is very powerful; this section is meant to provide a taste of a few things that can be done with it. If you need more detailed help, you can always send me email.

There are a number of different types of objects that ZipIt defines to make scripting easier:

• a window is synonymous with a document, and refers to an open zip archive
• an entry refers to a file or folder within a zip archive
• a file refers to a file within a zip archive
• a folder refers to a folder within a zip archive

Entries, files, and folders are numbered separately, so if you have an archive that consists of a file and then a folder, folder 1 will refer to the folder, file 1 will refer to the file, and entry 1 will refer to the file, and entry 2 will refer to the folder. Also, entries, files, and folders are numbered relative to the current window unless you specify otherwise. File 1, in other words, refers to the first file in the window, not the first file in the archive. To see the significance of this, suppose you open an archive with a folder in it. You then open the folder. File 1 now refers to the first file in the folder that you just opened - in other words, it changed, now that you've opened a folder. Absolute references are possible by the use of the "root" property of a window or document. To refer to the first file in the archive, you can refer to "file 1 of root of window 1". The root is the topmost level of the archive. Whenever ZipIt returns an object reference, it always refers to it from the root to avoid confusion.

Each object has elements and properties, which are listed fully in ZipIt's dictionary (choose Open Dictionary in Script Editor to see ZipIt's AppleScript dictionary). Windows and documents contain entries, files, and folders as elements, so you can refer to a file by saying "file 1 of window 1". You can also use names, e.g., "file "test" of window "archive.zip"".

To zip files:

```
tell application "ZipIt"

activate

set file_to_zip to (choose file)

set zip_archive to (make new window at the beginning)

add file_to_zip to zip_archive

compress zip_archive to file "Hard disk:Desktop folder:archive.zip"
```

end tell

The variable file_to_zip, above, can also be a list of files or folders.

To unzip an archive:

tell application "ZipIt"

activate

set archive_to_unzip to (choose file)

open archive_to_unzip

extract every entry in window 1 into file "Hard disk:desktop folder:" -- Note the use   of the 'file'
keyword here
end tell

To create a new folder, say "make new folder at end of window 1 with properties name:"foldername"". As an alternative to the "add" command, if you want more control (but less intuitiveness), you can write "make new file at end of window 1 with properites disk file: file "whatever"". You can naturally also specify other properties in the make command.

If you have further questions about scripting, the ZipIt homepage may have an answer to your question (see the introductory section for the URL). Otherwise, feel free to send me email.