

# StuffIt

Written by & ©1988 by Raymond Lau  
Portions ©1986. THINK Technologies, Inc.

Special thanks to David Schenfeld, Allan Weber, Richard Outerbridge.

version 1.30 of January 2, 1988

**NOTICE: StuffIt will only work with System 3.2 or later. System 3.2 is compatible with all Macs with 512k or more memory.**

(This is the most common “bug” report - a crash with ID 17...)

## Special Note on Updates:

I will mostly be depending on BBSs, timesharing systems and users' groups as the primary distribution media for StuffIt updates. Unless you include the \$18 plus disk & SASE or \$20 (for disk and mailing), I will not send you the latest version. Please tell me which version you already have so I won't send you a duplicate! If you do not find StuffIt useful, but would still like the latest version for decompression, send a disk and SASE - **no \$2 checks**. At any point in the future, if you'd like me to send you the latest version or maybe something new I've done, send a disk & SASE...

## Description and Introduction

Programs of increasing functionality and complexity are being introduced in the Macintosh world constantly. Many of these programs are large and/or consist of more than one file.

With programs consisting of multiple files, a user might forget to download an essential part of a program and is subsequently faced with anguish and confusion as he tries to reason out what went wrong.

With programs of a large size, the time needed to transfer a file is noticeable.

Clearly, there exists a need for utilities which allow for the grouping of multiple files into one (“archiving”) and the compression of file(s) to reduce download time and hassle.

StuffIt is one such program.

StuffIt is based in part on the IBM program Arc, in part on the UNIX program Compress and in part on my own evaluation of the needs of the Mac user. It is appreciably faster than PackIt. It is also more efficient than PackIt.

By the way, StuffIt was written with THINK Technologies' LightspeedC compiler, which works like a charm under MultiFinder!

## Distribution/Usage

StuffIt may be distributed as long as it is distributed for free. A normal duplication/usage time charge may be made as long as the rate charged for StuffIt's distribution is the same as that charged for other public domain, freeware and/or shareware programs.

StuffIt may NOT be distributed with any commercial product without my permission. If I feel that its distribution would be of a mutual benefit, I would allow it for free. However, if for example, the program is used to distribute sales demo files, I would have to ask for a time licensing fee or a per copy fee.

To encourage the use of StuffIt, I have devised the following deal:

You may use StuffIt for de-archiving/decompressing files for free.

You may try out StuffIt's compression/archiving and maintenance abilities for 15 days.

Afterwards, if you decide to continue using StuffIt to compress/archive files and otherwise maintain archives, you must register your copy for the shareware price of \$18. Please include your name, address, the full version number (eg: 1.20) and if you will, where you obtained StuffIt from. NOTE: I will mostly be depending on BBSs, timesharing systems and users' groups as the primary distribution media for StuffIt updates. Unless you include a disk & SASE with the \$18 or \$20 (extra \$2 for disk and mailing), I will **not** send you the latest version. Please tell me which version you already have so I won't send you a duplicate! If you do not find StuffIt useful, but would still like the latest version for decompression, send a disk and SASE - **no \$2 checks**. At any point in the future, if you send me a disk & SASE, I will send you a later version of StuffIt if available or anything else I might have written.

Basically what I am saying is, if you don't like it, don't use it for archiving/compressing. But don't throw it away either. Others may like it and may use it to prepare files for distribution. It wouldn't be fair if you were unable to use these files. So, you may continue using StuffIt to extract files from StuffIt archives. I think these conditions are fair.

Please feel free to modify your version as long as you do NOT distribute a modified version. You may personalize SIT in terms of adding DAs, changing some of the resources, etc... but I will reiterate my behest again, please do not distribute a changed version.

I may be reached at:

GEnie: RayLau

UseNet: raylau@dasys1.UUCP

MacNET: RayLau

CIS: 76174,2617

Delphi: RaymondLau

US Mail & registrations of \$15:

Raymond Lau  
100-04 70 Ave.

Forest Hills, N.Y. 11375

I need YOUR support to make this method of distribution work. Don't procrastinate. Don't have that "someone else can support it" attitude. Get involved!

## Operation

Operation is very straight forward. First a few simple definitions. An archive, particularly a StuffIt archive, is one or more files grouped together for any purpose, including storage, transmission, et cetera. A file is any file on a volume accessible by your Macintosh computer. An entry is any file compressed or uncompressed, which is stored in an archive.

### Working with a file archive:

Select New StuffIt Archive from the file menu (or press command-N). Enter a name for the archive and hit return or click on new. NOTE: It is suggested that all StuffIt archives end in ".sit". What this basically does is create an empty archive with no files in it. The free space on next volume button allows you to cycle through all mounted volumes and view the number of kilobytes (rounded to nearest k) free on the volume name specified at the bottom.

To add files to the archive, click on the add button or press command-A. Select the files you wish to add in the standard manner. If you want to add **entire folders** at once, select a folder, click on the **Add** button OR press either command-enter or command-return. Clicking on the open button will display the contents of the folder. When a file is selected, both the add and open buttons function identically. NOTE: StuffIt will now add **all** files in the folder itself and any subfolders. Adding to an archive attempts to compress the file and if the compression proved effective, the compressed entry will be stored in the archive. The Try ... radio buttons are explained later under Compression Methods.

If you're adding many files at once, StuffIt includes a feature which will allow you to select up to 20 files and add all the files in one shot. Click Multiple Add (or cmd-M). You will be presented with the standard add file dialog box. Select the files as you normally would, one after the other. You will notice that your selections will be added to a list. Click cancel when done. Now you can either delete files from the list (select files, click delete or cmd-d), add [more files] to list if you haven't filled the 20 file capacity (cmd-a does the same thing), cancel your entire list (cmd-. is the keyboard equivalent) or archive the list (return is the magic key sequence). If you select Archive, all the files on the list will be compressed as if you had added them one by one. This comes in handy when you have a group of large files to add and don't want to wait around. So, go have a cup of coffee while you're waiting!

Now that your files are in the archive, you may close the archive and move on to the next one. Or you may decide to delete a few files or extract a few. As you see (or will see), if there are any files in an archive and that archive is opened, you see a list of files contained in the archive.

Each entry consists of the file name, the type, the creator, the decompressed size, and the % saved by compression. For more detailed information, select the entry and click Info (or cmd-I).

Well, supposing that you wish to delete or extract files from an archive. If that archive isn't already open, close the current archive, if any, and then select Open StuffIt Archive from the file menu. Or press command-o.

To extract a file/files, select the file(s) you wish to extract. If you wish to select more than one file, shift click to select a range or command click to select a group of non-continuous entries. If you wish to quickly select all the files in an archive, select Select All from the edit menu or type command-S. Now, click extract, or press command-e. You will be prompted as to where to save each file. You will also be given a suggested name for each file. The size of the file being extracted at the moment is mentioned for your convenience. The free space on next volume button works in the usual manner. Click Save to save the extracted file to disk. You may also select Save All which has the same effect as clicking save for each selected file and saving them all in the same location. As an added convenience, you can tell the combined size of all selected and not yet extracted files by looking at the display towards the lower left hand corner of the screen. If there isn't enough room on the destination volume to save a file, you will be told such and that file will be skipped.

If you used Save All, the default suggested names will be used. Should a file name conflict arise, you will be prompted for a new name.

### \*\*\*Tip/Trick/Hint

To quickly extract all files from an archive or archives, select the archive or archives from the Finder. While StuffIt is loading, hold down the shift key. This has the same effect as selecting all the files, extracting them and "save all"ing them in the same folder as the archive. Again, you will be prompted to resolve file name conflicts should any arise.

**Also!** A new feature since 1.30 - after selecting Open StuffIt Archive..., if you hold down the shift key as you double click on the archive to open or press return - and **keep holding down the shift key**, it will select all the entries for you and auto-extract them to the same folder as the original archive!

As noted in the extract dialog box, there are a few shortcuts you may employ from the keyboard. Command-return is save all, command-period is skip and shift command-period is cancel.

If you wish to PERMANENTLY delete entries from an archive, select the files and click Delete, or press command-D. You will be warned that the entries will be gone forever, at which time you may continue or cancel. To bypass the dialog, hold down the option key when you press command-D or when you click on delete.

You can also get some information on entries in the archive. Select the entries you wish to query and click Info or press command-I. You will be told of its file name, compressed and uncompressed sizes, compression method if any, type, creator, creation date, modification date and if it can be found, the name of the application which created the file.

You may rename an entry in an archive by clicking on the Rename button or clicking command-R. Just enter its new name and click the appropriate button or press return.

If you're archiving files for storage or transmission, it may help to have a report of the archive. StuffIt will create a text file detailing the contents of the archive. This file may be later printed or kept on disk for use as reference.

**New in 1.30 and later!** The report's fields are now separated by tabs for import into spreadsheets and databases. I've tried it with Excel and it works. To view as a normal text file, please use a tab with of 8 or 16 characters in your text editor for best results.

One last note: The computation of all the sizes, for report, for the list of archive contents and for the info display do not take the size of the file headers into account.

## Archive Comments

New in 1.30 or later - You can store comments along with the archive by selecting Comment... from the file menu. It will display whatever comments currently exist and will let you edit/delete/alter the comments.

## Note on Modification Dates

If you would like the modification date restored along with your file, check the Restore Modification Date option under the Options menu. If you would not like it restored, uncheck it. There's a reason why you might not want it restored. Some hard disk back up programs rely on the modification date for incremental back ups, so if you restore it, the file might not be backed up! I'll leave this point for you to explore...

## Compression Employed

### Basic/RLE

StuffIt chooses from 4 compression methods. No compression, compression of repeated character runs and Lempel-Ziv compression (at 14 bits) and Huffman. It should also be noted that the two forks are compressed separately. Compression of repeated characters is efficient mainly for small files which might have many spaces or the like. This method will be shut off for all files larger than a cut-off point of 25K as it tends to be inefficient for larger files. For such larger files, LZW proves to be much more effective.

### LemepI-Ziv

Lempel-Ziv is a high performance technique which basically encodes repeated strings. I have chosen the 14 bit version because I have found that going to 15 or 16 bits increases memory requirements dramatically and required complicated indexing to the data structures used, thus slowing operations down drastically.

There are some cases where LZW falls short. Notable examples include MS Word 3.01 and MS Word's Main Dictionary, sound files, & graphics bitmap files.

## **Huffman**

To make up for where LZW leaves off, version 1.10 and above include the option to attempt Huffman compression of the fork. With the addition of Huffman encoding, the "worse case scenario" as it relates to comparisons with PackIt II/III is very promising: At most, a SIT entry will be approx. 75 bytes larger for a one fork file, and slightly more for a two fork file. For the few bytes, you get faster decompression, the ability to extract arbitrarily, to add to an archive, to delete from an archive, etc.

### **\*\*\*Selecting Algorithm**

Repeat packing (RLE) is ALWAYS attempted for files under 25k and NEVER attempted for files 25K and over. (Reasons given under discussion of LZW and RLE).

To select between LZW and Huffman, you have the following options:

Try BOTH-The slowest option, but you will always get an optimal result. If you're patient, or have a relatively fast machine to offset the speed degradation, choose this option.

Try LZW-Huffman will not be attempted. This option will probably be what you'll be using most of the time.

Try Huffman-LZW will not be attempted. This option works best for sound files, files with much bit mapped graphics, some stacks and some dictionaries (spelling).

These options are available under the Options menu as well as the add file selection box.

There is a reason why I don't have SIT choose the algorithm based on the file type. That would work well for some files, but how can SIT determine the graphical content of say a stack? I have had a Hypercard stack compress 41% under LZW/25% under Huffman. Then another stack wouldn't compress under LZW and compressed 11% under Huffman. So I've decided to leave the choice to you...

## **Data Integrity Assurance**

StuffIt maintains a CRC-16 of the data fork and the resource fork as well as the file hdr. If either fork is damaged, you will be told so. If the hdr is damaged, you will be told such and also that the file may be bad. There's no sure way for StuffIt to tell, since the CRC-16 of the two forks are stored in the header! By the way, the main archive header doesn't have a CRC.

## **Built-in Utilities**

Besides the UnPack utility, which is described later under Compatibility with PackIt II/III, StuffIt has two other built in utilities to make life easier for you.

## **BinHex**

The first is a BinHex4 .Hqx encoder/decoder. If you are unfamiliar with BinHex, then odds are that you won't need to ever use the utility. But if you do, selecting Encode BinHex file will allow you to select the file to be encoded and then the name and location to save the encoded file in. The encoded file will be fully compatible with BinHex4 and BinHex5. (For those techy BinHex fans, note that StuffIt will not do RLE encoding as BinHex does) To decode a .Hqx file, select Decode BinHex file, select the file to decode and then name the decoded file and specify where to save it.

## **Segmenter/Joiner**

Another feature is a built-in file segmenter/joiner. This utility allows you to divide a large file up into smaller pieces, so that they'd fit on a disk or so that you won't have to send a large file at once...

The first thing to do is select Set Segment Size... Here, use the scroll bar to determine the size of each "segment" of the divided file. The size is changeable in 10k increments from 10k to 3200k. Bear in mind that a 400k disk only has 391k free after formatting... **Make sure you have enough space to save the segments!**

Next, select Segment File... Specify the file you wish to segment. And then select a name and location to save each segment of the file.

To Join files, select Join Files... Find the **first** segment of the file to be joined. Select a location to save the joined file. **Make sure you have enough space to save the joined file!** As the join operation progresses, you will be prompted to find the next segment of the file, and then the next, so forth...

## **Compatibility with PackIt II/III**

Currently, StuffIt offers compatibility with PackIt II/III by allowing the user to unpack .pit files and to incorporate the unpacked file(s) into a StuffIt archive. NOTE: StuffIt doesn't support PackIt III's compression method.

To unpack/convert files, select UnPack... from the PackIt menu. Select the .pit file you wish to unpack. As each packed file is reached for retrieval, you will be prompted with a suggested file name. At this point, you may Save the file (or Save All to save all files from this point on), Convert the file (or Convert All to convert all files from this point on), Skip the file, or Cancel the UnPacking.

NOTES: If you are converting (a) file(s), they will be UnPacked and then added to the currently opened StuffIt archive. If an archive isn't opened, the convert buttons will be dimmed. Conversion uses the active System folder. Make sure there is enough room in it to accomodate the file(s) you are converting. There's no quick way to skip through a .pit file. Each entry must be decompressed to reach the next. That was one of PackIt's many faults...

## Future Changes/Thoughts/Etc.

De-Arc ability - this is the big one I've been talking about and have been neglecting. Some day I will get around to it. I keep putting it off bec. it represents a major addition and it requires the assistance of someone with a PC.

Continuation files - this is a tough one. I do not have an idea yet as to how I will go about this. But be forewarned that if it is added, using this feature will freeze the archive - i.e.: no future additions or deletions.

Encryption - I might tackle this next...

## Credits (Thanks to...)

Original idea of implementing a Lempel-Ziv archive utility was that of David Schenfeld. Sources for Arc (of System Enhancements) and Compress (of UNIX) provided by David Schenfeld.

UnPacking code from Allan Weber—obtained through David Schenfeld.

Access to Compress,UnPack code ultimately provided by the operators of Big Electric Cat public access Usenet node.

Icon, thanks to J.W. McGuire.

Testing, thanks to Andrew Lih, Leonard Rosenthal, Anthony Valero and others.

Assistance with assembly from Richard Outerbridge was also most appreciated!

## StuffIt Archive Format/Notes

Familiarity with Arc, and preferably Compress too, is assumed.

The repeat character compression is the same as the RLE algorithm used by Arc (hdr vers 3). The Lempel-Ziv compression is the same as that of Compress @ 14 bits or a modified version of Arc's Squash (hdr vers 9)...@ 14 bits with hash table size @ 18013 entries. Huffman compression uses a strictly Huffman encoding scheme (no RLE, no scaling of codes to force them w/in a certain length, etc. ). The fork entry for a Huffman compressed fork is a dictionary followed by the compressed fork itself. The dictionary is identical to that of PackIt II/III. Please refer to appropriate documentation for more information. CRC is calculate with the table lookup function used in Arc (which is the UNIX/PC standard CRC16. It is **not** quite CCITT!) Now that you have the algorithms, the file format and header formats as declared in C are:

```
/* Stuffit.h: contains declarations for SIT headers */
```

```
typedef struct                                /* 22 bytes */
{
    OStype signature;                          /* = 'SIT!' -- for verification */
    unsigned int   numFiles;                    /* number of files in archive */
    unsigned long  arcLength;                   /* length of entire archive incl.
                                                hdr. -- for verification */
    OStype signature2;                          /* = 'rLau' -- for verification */
    unsigned char  version;                     /* version number */
    char reserved[7];
} sitHdr;
```

```
typedef struct                                /* 112 bytes */
```

**StuffIt** version 1.30 of January 2, 1988

pg.8



```

{   unsigned char  compRMethod;      /* rsrc fork compression method */
    unsigned char  compDMethod;      /* data fork compression method */
    unsigned char  fName[64];        /* a STR63 */
    OSType fType;      /* file type */
    OSType fCreator;   /* er... */
    int FndrFlags;     /* copy of Finder flags. For our
                        purposes, we can clear:
                        busy,onDesk */

    unsigned long  creationDate;
    unsigned long  modDate;      /* restored? */
    unsigned long  rsrcLength;    /* decompressed lengths */
    unsigned long  dataLength;
    unsigned long  compRLength;   /* compressed lengths */
    unsigned long  compDLength;
    int rsrcCRC;      /* crc of rsrc fork */
    int dataCRC;      /* crc of data fork */
    char reserved[6]; /* invalid data for now */
    int hdrCRC;      /* crc of file header */
} fileHdr;

```

/\* file format is:

```

    sitArchiveHdr
        file1Hdr
            file1RsrcFork
            file1DataFork
        file2Hdr
            file2RsrcFork
            file2DataFork
        .
        .
        .
        fileNHdr
            fileNRsrcFork
            fileNDataFork
*/

```

/\* compression methods \*/

```

#define noComp 0    /* just read each byte and write it to archive */
#define rleComp 1   /* RLE compression */
#define lzwComp 2   /* LZW compression */
#define hufComp 3   /* Huffman compression */

```

/\* all other numbers are reserved \*/

Archive comments are stored as a pure text resource of type SitC and ID 0 with the archive. The length of the comment is the length of the resource.

The format of segmented files available by request.

.Hqx encoding is done without RLE in Stuffit's implementation.