

## UUTool 2.4

By Bernie Wieser © 1990 to 1995

Octavian Micro Development Inc.

ALL RIGHTS RESERVED

### Contents

- 1.0 Licensing Agreement
- 2.0 File Menu Features
  - 2.1 UUcode
  - 2.2 Text Segmentation
  - 2.3 Finding Files' Creator and Type
  - 2.4 Audio Files
  - 2.5 Preferences
- 3.0 Revisions Log
- 4.0 Trouble Shooting
  - 4.1 XXcode
- 5.0 How to Contact Me
- Credits

#### 1.0

##### Licensing Agreement

Octavian Micro Development hereby releases UUTool, a program for encoding and decoding UUENCODED format files, as 'freeware' on the following conditions. The user understands that UUTool is the property of Octavian Micro Development, and that Octavian Micro Development is not responsible for any damages incurred through the use of said tool. UUTool is not for commercial use without prior registration with Octavian Micro Development. UUTool must not be sold, or distributed on/in any media for profit (including mail order) without the prior consent of Octavian Micro Development. Use or distribution of said tool implies this agreement is understood and abided by.

The coding engine in UUTool is available for licensing. Please contact me for more details. If you would like to support UUTool's development, please send a donation to the address below.

#### 2.1

##### UUcode

The UUcode menu item from the File menu offers an implementation 'uuencode' and 'uudecode' binary to text file utilities for the Macintosh™. The purpose is to translate binary files into text for transmission over some electronic medium (i.e. electronic mail, usenet news, etc.). It is compatible with many UNIX™ implementations, with extra coding to support the two file fork Macintosh™ architecture.

UUTool currently supports multiple file decode on opening of the UUTool application only. From the Finder, select your target text files and select Open from the file menu, or drag the selected icons onto the UUTool icon.

#### 2.2 Text Segmentation

The Segment menu item allows for the multiple splitting and joining of text files. The join Læ available from the Segmentation preferences allows you to change headers for decoded file name, segment number, and start and end cut lines. These exist in a queue and their weighting is determined by the queue element number (which you cannot change in 2.3). All split input files are pre-scanned before joining. Name and parts are guessed at until a cut line is reached. Then all the parts which seem contiguous are concatenated into an output file.

Split will split text files by size or by number of lines. Header information is included in both cut lines which are inserted before and after the segmented text. Splitting is a two stage process. The source text file is split into its parts. Then the parts are opened to make their header information textual.

UUTool only understands the checksum its own headers. This checksum is the result of adding all the text characters in the segment of the source file, not including the headers. (Checksum is not implemented in 2.3)

Join is a slow process. It also is not fool-proof. You might have to use a text editor to join the parts anyway.

### 2.3 Finding Files' Creator and Type

The Show Creator and Type menu item allows you to easily hunt down the creator and types of files. Select a target file with the standard file dialog to see it's creator and type. If the file is a document, click the Find Creator button to show the file types supported by the creator application.

Known bugs include confusion on machines with multiple desktop files with Find Creator.

### 2.4 Audio Files

UUTool supports the translation of 8 bit audio files to AIFF or Macintosh™ linear format, as well as AIFF to ULAW or linear audio files. Sampling rate is maintained when using AIFF format. If the AIFF file is corrupt, then use linear format instead and set your sampling rate to 7 kHz with whatever sound program you're using for playback.

## 2.5

### Preferences

There are three options currently under the preferences menu: Segmentation, UUcode, and Save Prefs. Save Prefs. will save all preference settings.

#### 2.5.1 Segmentation preferences

The Segmentation preferences currently support splitting a file by line count or by size. Optionally you can add a checksum.

Know your host's text limits. AppleLink has a 32K limit. CompuServe has a 50K limit.

The Join DB button gets you to the header database. There are four fields for file name, part number, and cut begin and end lines. These lines are compared with lines in the source file. There are four join variables to be set. Each variable begins with a 'ð' symbol. ðchksm allows you to denote where an 8 character hexadecimal checksum of the segmented data. ðfile allows you to denote where the original file name would be.

$\partial$ part and  $\partial$ total allow you to denote which part of how many in the segment you are working on. In addition, the '...' character means 'accept all input until I match the character following ...'. The '\r' character means match a carriage return. The 'Use It' check box denotes whether or not you want that header record to be used in the search.

Examples:

"Subject:  $\partial$ file ...[ $\partial$ part/ $\partial$ total" is valid

"Subect:... $\partial$ file" is invalid, since the character after ... is  $\partial$

## 2.5.2 UUcode preferences

### Macintosh™ Preferences

By default Macintosh™ files are encoded in their entirety. UUTool's decode can see Macintosh™ formatted files and restore them to their original condition. The data fork of a file is the first part in the encoded text file, bounded by a 'begin' and 'end' statement. This is followed by the resource fork, and finally finder information (i.e. the file's icon, etc.) When the check box for data fork only is on, only the data fork of a file is encoded or decoded. Either way, on a UNIX™ system only the data fork will be decoded.

The "use original coding" check box will use the standard uu\*\*coding method, necessary for many PC implementations. I suggest not using original coding when using UNIX™ machines, since some electronic mailers reformat white space.

The PC, UNIX™, and Macintosh™ radio buttons allow for different ways of encoding a file's name. The PC button on means the name will be truncated to eight characters, an extension, and no special characters will be used. The UNIX™ button on means no special characters will be used. The Macintosh™ button on means anything goes! Be careful about characters in file names. The PC and UNIX™ buttons will replace dangerous character (like '/' and '\') with underscores. Extended characters will be replaced by the hash symbol ('#').

Default creator and file type are only used if UUTool does not find Finder information, or can't guess at it. With the SetType check box on, UUTool will scan through the SetType signature database and try find a match.

### UNIX Preferences

UNIX™ permission defines the file's permission modes if the file is to be decoded on a UNIX™ machine. For example, '0700' stands for read/write/execute for owner only. This is ignored on the Mac side.

You can put whatever text you like before the first 'begin' statement or after the last 'end' statement (except those tokens must start on a new-line!) UUTool should be able to decode files with headers. Try not to fiddle with information between those tokens.

### SetTypes

You can specify UUTool's guess for a file's creator and type with the SetType button. This involves knowing something about the internal structure of your decoded files. Most reasonable programers have their data files identify themselves with a static signature somewhere in the file. If you provide a signature of less than 9 bytes, and an offset, UUTool will set the suggested creator and type of the decoded output file if there

is a match. I recommend using a disk editor to obtain this information.

### 3.0

#### Revision Log

v1.1 - fixed drawing into a NIL window brought about by file errors.

v1.2 - fixed SysBeep problem under finder (not having enough memory to be loaded back in).

v1.2.2 - fixed some port problems

v1.2.3 - fixed an addressing problem for 68000 machines

v2.0b - stream i/o completely removed with faster new engine (UENG)

- added encode/decode status (allowed with new engine)

- no longer uses all memory but works with 110K

- no longer supports CR/LF encoding (CR only)

v2.0b1 - corrected dialogue problem under system 7.0

- added app. file open support

2.0.1 - added 'mstr' support

- added keyboard shortcuts to prefs. as well as their feedback

- uses UENG 1.0.1 which allows for variable buffers (still uses old sizes)

- removed debug traps in UENG

- all code is 32 bit clean and 68000 compatible (I hope :-)

2.0.2 - fixed ignore EOF in uudecode UENG

- added user abort to encode/decode

2.0.3 - uses UENG 1.0.3, nicer abort

2.0.4 - added original coding option

2.0.5 - moved most strings into resources for localization purposes

2.0.6 - added on-line documentation and print

- changed GetNextEvents to WaitNextEvents (no WNE check though)

- added some EventAvails (so now background tasks get time)

2.0.7 - converted to Think C 5.0.2 by Bob Abeles

2.0.8 - slightly optimized uu-engine 1.0.4

- supports balloon help

2.0.9 - fast new begin scan

- work around system 7.0 SetFPos bugs (uu-engine 1.0.5)

2.0.10 - support for audio files

2.1 - removed Balloon help, Split, and other unused things

- added file error catching to sound stuff

- changed UNIX™ encoding table to ignore ,

- more optimized and bug fix uu-engine 1.0.6

2.1.1 - another bug fix to the uu-engine 1.0.7

- bug fixes regarding error messages

2.1.2 - fixed bug regarding PC file name's length

- added Show Creator/Type and GuessType support

- changed decode to only decode data fork if "data fork only" option is set

2.1.2.1 - fixed decoded file name bug

2.1.2.1 - fixed a DIV0 on the status bar when encoding an empty file

2.2 - unloaded Scrap to fix memory problems

- added Split support
- moved SetType configuration to UUcoding preferences
- rearranged all the menus
- added Gestalt calls to check system config. and fixed error messages
- 2.2.1 - changed encoding file name to use one of three formats, and changed decoding to take everything to EOL (end of line).
- 2.3 - added multiple decode files
- released join beta
- 2.3.1 - hmmm... can't attach an invisible list to a NIL port; try the window manager's
- so, Scrappy, I unloaded 'u' case 'u' gave me trouble, now you're zero'd
- 2.3.2 - fixed a bug to be system 6 compat. in SFGetFiles
- 2.3.3 - updated and re-compiled everything
- 2.3.4 - fixed the decode problem
- added a flushvol
- moved prefs. from data fork to resource fork
- 2.4 - accelerated for Power Macintosh!
- supports core apple events
- sped up split and join
- updated Balloon help

## 4.0

### Trouble Shooting

- If you get a -108 error, allocate UUTool more memory from the Finder. This error is usually associated with insufficient application memory for the clipboard.
- Don't run UUTool from a locked disk, or have the application software locked.
- the Find creator feature only works for the boot desk-top DB.

If you don't like the UUcoding translate tables I provide with UUTool, you can change them. They exist in HEXA resources 128 and 129.

### 4.1 XXCode

This program is a uu\*\*coder, not an xx\*\*coder. XX is a strain of uu\*\*code which allows users of computers which use EBCDIC character coding (mainly IBM mainframes) to translate binary files into text files and visa versa. I have modified my UNIX translate table to not use the " or " characters, but the '^' character presents a bit of a problem. '^' should become '~' for xx\*\*coding. Until such time as someone funds me to implement an xx\*\*coder, you can use a text editor and globally replace '^' with '~' and it should work. UUTool will NOT decode these files properly!

If you find any bugs or have any comments, feel free to write me a letter (address below.)

## 5.0

### How to Contact Me

Octavian Micro Development is a software/hardware development company. I

(Bernie) wrote UUTool in 1990 when I was an undergrad, but I am a real software engineer now and know how to write real Macintosh applications. Version 2.4 is the last version of UUTool, so enjoy it. You can buy the source code for \$50 US.

My address is:

Octavian Micro Development Inc.  
4115 Brisebois Drive N.W.  
Calgary, Ab., Canada  
T2L-2E8

Office Phone # (403)289-5986  
e-mail [Octavian.bsw@eworld.com](mailto:Octavian.bsw@eworld.com)

Credits:

I'd like to thank everyone who has sent me positive comments. And to those who sent negative comments, bite me, its free.

Macintosh™ is a registered trademark of Apple Computer Inc.  
UNIX is a registered trademark of AT&T Information Systems.