

UUhx

Adam Dawes

Copyright © CopyrightÂ©1995 Adam Dawes

COLLABORATORS

	<i>TITLE :</i> UUhx		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Adam Dawes	August 4, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	UUhx	1
1.1	UUhx Documentation	1
1.2	What is UUhx?	1
1.3	Why use UUhx?	2
1.4	System requirements	2
1.5	Using UUhx	2
1.6	Legal stuff	4
1.7	History	4
1.8	Things to do	5
1.9	Contacting the author	5

Chapter 1

UUhx

1.1 UUhx Documentation

UUhx v1.01

By Adam Dawes
24th June, 1995

What is UUhx?
Why use UUhx?
System Requirements
Usage
Legal Stuff
History
To Do
Contacting the author

1.2 What is UUhx?

If you've ever tried to send a binary file over (for example) an electronic mail network, you'll no doubt have realised that there are a few problems. Your mail system will completely mangle any binary files that you try to send, and the result will be a corrupt file arriving at the other end.

UUhx is a utility that will convert any binary files to a purely text file, or will convert the text file back to a binary file. It uses a standard known as 'UUencoding', and programs to en-/decode UUencoded files can be found on just about every computer platform in use today (UNIX, VMS, PCs, etc.).

UUhx is compatible with all these other UUencoding/decoding utilities, so it should be very easy for you to send and receive UUencoded files to/from other computer platforms using UUhx.

1.3 Why use UUhx?

UUhx has several advantages over the other UUcoding programs available for the Amiga:

- Speed

UUhx is much faster than other UUCoders. For example, over 30% faster at both encoding and decoding than UUXT.

- Full archiver support

UUhx is capable of automatically archiving files before UUencoding, and of automatically unarchiving files after decoding. Archivers currently supported are Stefan Boberg's LhA and Jonathan Forbes' LZX. Complete archiver configurability will be added in the next version of UUhx so that *any* archiver can be used for both compressing and uncompressing files.

- Simple and flexible parameters.

Using/changing archivers is very straightforward. Wildcards fully supported for archivers.

- Freely distributable.

Does not add stupid shareware messages to encoded files.

1.4 System requirements

UUhx should work on any Amiga running kickstart 2 or greater.

1.5 Using UUhx

When run with no parameters, UUhx will print the following usage summary to the screen:

```
Usage: UUhx a <uuefile> <sourcefile> ..... encode a file
       UUhx x <uuefile> ..... decode a file
       UUhx l <uuefile> <archiver> <sourcefiles> ..... archive/encode
       UUhx u <uuefile> <archiver> ..... archive/decode
```

The basic options are as follows:

'a' (encode a file)

This option instructs UUhx to take a binary file (specified as 'sourcefile' in the usage summary) and convert it to a UUencoded text file ('uuefile').

For example:

```
UUhx a AmigaPic.uue ram:Amiga.iff
```

This will convert the binary file 'ram:Amiga.iff' to the UUencoded file 'AmigaPic.uue', in the current directory.

'x' (decode a file)

The 'x' option will convert a UUencoded file back to its original binary form.

For example:

```
UUhx x MAIL:Inbound/AmigaPic.uue
```

This will convert the AmigaPic.uue file in MAIL:Inbound/ back to its original form. The original filename ('Amiga.iff', in this case) is contained within the UUencoded file, so there's no need for you to specify it when extracting the binary file.

'l' (archive and encode file(s))

This option is similar to the 'a' (encode) option, but allows you to use an archiver of your choice to compress files to be encoded before the UUencoding process takes place. You may use any supported archiver for the compression (currently LhA and LZX are supported).

Some examples:

```
UUhx l Picture.uue LhA AmigaLogo.iff
```

This will invoke LhA to compress the AmigaLogo.iff file, and then will UUencode the resulting archive. Note that the archive is given the same name as the resulting UUE file, but without a .uue extension if one is present (so in this case the archive will be called 'Picture.lha'). Also, the archive is created in T: and deleted after it has been encoded, so you never need worry about cleaning up all the archives after they have been encoded.

```
UUhx l Pictures.uue LZX ram:Pic1.iff ram:Pic2.iff ram:Pic3.iff
```

This will invoke LZX to compress the three files specified to the archive 'Pictures.LZX', and will then UUencode the archive to 'Pictures.uue'.

```
UUhx l TestFile.uue LhA ~(*.info) ENV:
```

This invokes LhA to archive all files in the current directory except for icons and all of the ENV: directory to 'TestFile.lha', and to then Uuencode the archive to TestFile.uue.

'u' (decode and unarchive file(s))

The 'u' option is the reverse of the 'l' option. It first decodes the specified Uue file, and then uses your selected archiver to decompress the files contained within the archive. Note that this option *does* leave the resulting archive in the directory (instead of deleting it, as is the case for the 'l' option).

Example:

```
UUhx u MyFile.uue lzx
```

This will first decode MyFile.uue, and then UnLZX the resulting archive.

1.6 Legal stuff

I won't accept responsibility for any damage done to your system or data lost, directly or indirectly, as a result from using this program or any of its associated files. You use the program entirely at your own risk. Of course if you *do* experience problems then I'll do what I can to sort them out, and please let me know so that I can try to cure them in a future release.

UUhx and its associated files are not public domain. They may be distributed freely as long as no unreasonable charge is imposed. They may not be included within any commercial package without express written permission from the author; the exceptions from this are the AmiNet CDs and Fred Fish's collections. UUhx may only be distributed if all files contained within the original archive are present.

1.7 History

v1.0 (39.0) -- 19/6/95

Initial release.

v1.01 (39.1) -- 24/6/95

UUdecoding now 40% faster than v1.0

LhA and LZX (de)archiving fully supported.

Memory deallocation problems fixed.

1.8 Things to do

To do:

Graphical User Interface option.

Full archiver configurability, so that *any* archiver can be used for encoding/decoding.

Automatic archiver detection for decoding.

Splitting encoded files to multiple smaller files for posting to systems that restrict the size of mail/news messages.

Decoding multiple split files to a single binary file.

Output of UUdecode/dearchiving to selectable directory, not just the current directory as at present.

1.9 Contacting the author

If you like UUhx, have found a bug, or have suggestions for its future, please contact me! I can be contacted via electronic mail or snail mail. I can't promise to reply quickly to snail mail, but I will always reply to email messages. I won't complain if anyone decides to send money! :)

InterNet:

ad32@brighton.ac.uk

FidoNet:

Adam Dawes@2:441/93.5

Snail Mail:

Adam Dawes
47 Friar Road
Brighton
BN1 6NH
England

UUhx is a "BeachWare" product
from the Beachy Head Amiga BBS, Eastbourne, UK
2:441/93@FidoNet +44 (0)1323 520999
7 days/24 hours 300-28800 bps 2 lines
