

# **LhA2LZX\_unreg**

Richard Burke

Copyright © 1998 Richard Burke

---

**COLLABORATORS**

|               |                                 |              |                  |
|---------------|---------------------------------|--------------|------------------|
|               | <i>TITLE :</i><br>LhA2LZX_unreg |              |                  |
| <i>ACTION</i> | <i>NAME</i>                     | <i>DATE</i>  | <i>SIGNATURE</i> |
| WRITTEN BY    | Richard Burke                   | July 7, 2022 |                  |

**REVISION HISTORY**

| NUMBER | DATE | DESCRIPTION | NAME |
|--------|------|-------------|------|
|        |      |             |      |

# Contents

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>LhA2LZX_unreg</b>                                  | <b>1</b> |
| 1.1      | LhA2LZX v2.7b AmigaGuide Documentation . . . . .      | 1        |
| 1.2      | Introduction . . . . .                                | 2        |
| 1.3      | How to use LhA2LZX . . . . .                          | 2        |
| 1.4      | Requirements - What you need to run LhA2LZX . . . . . | 3        |
| 1.5      | Creating AmigaDOS Scripts . . . . .                   | 4        |
| 1.6      | Arguments . . . . .                                   | 5        |
| 1.7      | Environment Variables . . . . .                       | 6        |
| 1.8      | Bugs in LhA2LZX . . . . .                             | 7        |
| 1.9      | Program History & Development . . . . .               | 9        |
| 1.10     | For the Future . . . . .                              | 14       |
| 1.11     | Legal Stuff - Disclaimer . . . . .                    | 14       |
| 1.12     | What I Used . . . . .                                 | 15       |
| 1.13     | Thanks to . . . . .                                   | 16       |
| 1.14     | frank . . . . .                                       | 16       |
| 1.15     | Contacting the Author . . . . .                       | 17       |
| 1.16     | Also by the Author . . . . .                          | 17       |
| 1.17     | Archiving . . . . .                                   | 18       |
| 1.18     | De-archiving . . . . .                                | 18       |
| 1.19     | The Command Line Interface . . . . .                  | 18       |
| 1.20     | LhA & LHZ . . . . .                                   | 19       |
| 1.21     | LZX . . . . .   | 19       |
| 1.22     | RequestChoice . . . . .                               | 19       |
| 1.23     | RequestFile . . . . .                                 | 20       |
| 1.24     | Index . . . . .                                       | 20       |

---

# Chapter 1

## LhA2LZX\_unreg

### 1.1 LhA2LZX v2.7b AmigaGuide Documentation

```
                LhA2LZX
~~~~~
The Archive Converter
© 1998 Richard Burke

Introduction
    What LhA2LZX does

Usage
    How to use LhA2LZX

Requirements
    What you need to run it

Writing AmigaDOS Scripts
    Some helpful hints

Bugs
    Problems in LhA2LZX

Program History
    How LhA2LZX developed

For the Future
    What should be in the next release?

Legal Stuff
    Disclaimer

What I Used
    ...in creating LhA2LZX

Thanks to...
    People who have helped me

Contacting the Author
```

---

Contacting me

Also By The Author  
Other scripts I have written

Index  
Run LhA2LZX

## 1.2 Introduction

Introduction

When

LhA  
came along, Amiga users were able to store files in one ( ←  
generally)

smaller file - an LhA  
archive  
. Then  
LZX

came along - allowing much better  
compression rates for some files, so much more space could be saved.

When I first got LZX I found it a lot of work converting my old LhA  
archives to the new LZX format via the

CLI

, and sometimes found that after

doing it that my original LhA files were smaller than the new LZX ones! Thus  
LhA2LZX was born. This script file allows the user to convert their archives  
from LhA to LZX or vice versa, or alternatively it will intelligently  
convert the archive to the smallest format! It is all done by requesters  
- so the user doesn't even need to touch the keyboard!

What makes it useful is that this program is an AmigaDOS

script

file, which

means you can load it into a text viewer/editor and see how it all works. I  
have commented many of the lines to help the user understand more what is  
happening, and to help in case you want to write your own front-ends to CLI  
programs, or to perform certain actions.

## 1.3 How to use LhA2LZX

Usage

- Double-click on the LhA2LZX\_v2.7b icon
- Select the FULL PATH of the directory that your original archive is in
- Select the archive
- Choose where a temporary directory should be made (to store the files of  
the archive whilst de-archiving). There should be at least 3-4 times  
the size of the original archive here

- Choose whether to convert to the other archive type (i.e. LhA to LZX or LZX to LhA) or simply to the smallest archive
- LhA2LZX will convert and save the new archive in place of the original

- LhA2LZX can be run from the CLI by simply typing

```
execute <path>/LhA2LZX_v2.7b
```

where <path> is the path of the directory it is in.

## 1.4 Requirements - What you need to run LhA2LZX

### Requirements

LhA2LZX requires several things to work:

```
[ BaseName      (1.3, arp component)  ]
  CD            (1.3 in C:, 2.x/3.x internal)
  Copy         (1.3/2.x/3.x in C:)
  Delete       (1.3/2.x/3.x in C:)
  Echo         (1.3 in C:, 2.x/3.x internal)
  Else         (1.3 in C:, 2.x/3.x internal)
  Endif        (1.3 in C:, 2.x/3.x internal)
  Eval         (1.3/2.x/3.x in C:)
  If           (1.3 in C:, 2.x/3.x internal)
  Lab          (1.3/2.x/3.x in C:)
  List         (1.3/2.x/3.x in C:)
  MakeDir      (1.3/2.x/3.x in C:)
  Quit        (1.3 in C:, 2.x/3.x internal)
  RequestChoice (3.x only, located in C:)
  RequestFile  (3.x only, located in C:)
  Resident    (1.3 in C:, 2.x/3.x internal)
  Skip        (1.3 in C:, 2.x/3.x internal)
  Which       (1.3/2.x/3.x in C:)
```

As indicated, you will already have these either in ROM or on your Workbench disks in the C: drawer (or already on your hard disk).

A hard disk is greatly recommended, and some spare RAM (depending on the size of the archives you are converting).

Both the

```
LhA
and
LZX
```

tools will also be needed - preferably in C:

[these are both available on the AmiNET, and from most PD libraries]

LhA can be found at util/arc/LhA\_e138.run and LZX is at util/arc/lzx121r1.lha  
Since

```
v2.6
```

the registered version of LZX is needed - and the registered keyfile is now freely available and part of the LZX distribution.

IconX will also be needed in C: if you want to double-click the icon to

start LhA2LZX.

From

v2.3  
+ of LhA2LZX, BaseName is no longer used.

From

v2.7a  
+, a 68020 or faster CPU is needed.

Versions of

RequestChoice  
&  
RequestFile  
for 2.x are available on the AmiNET.

Tested on:

A1200, 25Mhz '020 + FPU (Apollo Turbo 1220), 170 MB HD, 2/4 MB RAM  
A1200, 25Mhz '030 + FPU (Blizzard), 580 MB HD, 2/4 MB RAM  
A1200, 40Mhz '030 + FPU (Apollo 1230), 2.1 GB HD, 2/16 MB RAM

Untested on KS 1.x/2.x.

If you can test it on any of these, please mail me with the results :^)

## 1.5 Creating AmigaDOS Scripts

Creating AmigaDos Scripts

Read LhA2LZX

AmigaDOS scripts are very useful because everyone who has Workbench has the necessary software to write them. They are written in a language which is easy to learn and easy to understand, and can be used to do many things. The startup-sequence is the most well known of these scripts, and shows just what they can do. Programs can be called and started, and

arguments

can be

passed to them, which is essentially how LhA2LZX works -

LhA

and

LZX

are

called from the LhA2LZX script and the arguments that the user has selected are passed to them. An important aspect of LhA2LZX is the use of

Environment Variables

.

Scripts are written using a text editor (such as Ed), and do not need to be compiled, so you can simply run them from Workbench or the

CLI

(which is

great when you're writing them - just the click of a button to test it!).

Comments (anything written after a semi-colon ";") are ignored when the



script is running, so don't be afraid to comment your file so you clearly know what bit does what. They will not affect the actions of the script.

If you do not have an AmigaDOS manual, detailing what all the keywords do (there wasn't one in the A1200 distribution), there are many available in the Public Domain - the one I use is the fantastic

DOSMan

by Peter Bagnato

which details the usage of commands up to and including the new WB 3.1 ones, and gives examples of how to use them. Also get your hands on other script files such as the other ones I have done

, or

DiskSqueeze

, a DMS

alternative that crunches entire disks with LZX! That file helped greatly in creating LhA2LZX.

## 1.6 Arguments

Arguments

Arguments are the keywords that follow the program's name if started from the

CLI

, and tell the program (for example) which file(s) to use, any ↵  
special

command or options to use and so on.

For example, if you typed

LZX ?

in a CLI, you would get a list of the arguments you can use:

LZX [-<options>] <command> <archive> [<file>...] [<destdir>]

where options are any options (such as make filenames uppercase)

command is what action you are performing (e.g. archiving)

archive is the name of the archive e.g. to be saved

file is the name(s) of the file(s) to be e.g. archived

destdir is the name of the destination directory where the archive is  
e.g. to be saved.

e.g.

if you typed at a CLI

LZX -F a Test.lzx dh0:Games/#? dh1:

you would be archiving all (#?) the files in the dh0:Games/ directory, with a Fast progress display (-F) and the archive would be saved as Test.lzx in the dh1: directory.

## 1.7 Environment Variables

### Environment Variables

Note: in the following examples I have missed off the prefix "nl2z" of each variable so that it is easier to read, as of

v2.6a

Environment Variables are exceedingly useful in script files as they allow ↔

numbers or text to be saved to a small file in ENV: (in RAM:) and used by any program. What makes them so useful is that if a dollar sign (\$) precedes the filename when it is called, the contents of the file are used instead. So if for instance the text dh0:Games/ was saved as the file ENV:zxc, and you typed

```
cd $zxc
```

in a CLI, the current directory would change to dh0:Games/ - just as if you had typed

```
cd dh0:Games/
```

Here is a relevant part of LhA2LZX that shows this:

Click to find in LhA2LZX

```
if $suf EQ lha
  cd $drwl
  cd l2z
  $wlha x -a -F -M -x $ldfl
```

Previously in the script, List was used (found in C:) to get the suffix (all the letters after the final full stop in the filename) of the filename to determine if it was an LhA or LZX archive, and the suffix was saved as the file ENV:suf. So if the file is an LhA archive, the line would be the same as "if lha EQUALS lha". It would then cd to the directory the user had chosen to put the temporary storage directory l2z, and would then cd into l2z itself. Next it would call LhA (with the location of the LhA program saved as the file ENV:wlha) and extract all the files, preserving their attributes (-a) from the chosen archive (the filename of the archive is saved as ENV:ldfl), with a Fast progress display (-F), preserving the path names (-x) with no "autoshow" files (which automatically display themselves when LhA de-/archives them). So if LhA was in C: and the archive was called dh1:Test.lha (which would have been selected by the user via a requester), the final line would be the same as

```
C:LhA x -a -F -M -x dh1:Test.lha
```

Environment Variables can be used also to store numbers:

Click to find in LhA2LZX

```
eval >env:flmem $flsz * 4
```

More info

This line calls Eval (in C:), and tells it to multiply the ↔ size of the

original archive (found with List and saved as ENV:flsz) by 4, and to save the answer as ENV:flmem instead of printing it to screen. This is used in LhA2LZX to determine if the user's machine has enough memory to perform the

action of de-/archiving (LZX archives can de-archive sometimes to 3-4 times the size of the archive) if they have chosen RAM: to temporarily store the files of the archive, and warns them if there may be insufficient memory - note here that file redirection is used (the > sign), which enables user's choices to be saved in files rather than be printed to the screen.

When creating your own script files, it is a sound idea for the very first line to be

```
set echo on
```

as this will give a display of all the actions being performed, so you can see exactly what is happening and where things need to be changed to fix any problems. This can be deleted when the script is finished and everything works as you would want it to.

## 1.8 Bugs in LhA2LZX

### Problems

Apparently the clear screen command (echo "\*ec") doesn't work for KS 2.0, so I changed it back to "\*e[1;1H\*eJ" in

```
v2.6a
```

```
(this moves the cursor to column
```

```
1 row 1 and clears all that follows it).
```

When de-archiving, if it encounters an empty directory, LZX may display the error message

```
*** LZX: Error writing to disk; disk full?
```

but still continues to de-archive regardless. This seems to be a bug in LZX, and seems to have no effect on the process. When the files are archived by LhA seconds later, the empty directories are also archived as normal.

e.g. Here is a copy of the output from one of my tests:

```
LZX 1.21 (Evaluation) Archive/Extract utility - 68020/68030 Version.
Copyright © 1995 Data Compression Technologies. All rights reserved.
Commercial use of this unregistered program is prohibited.
```

```
Extracting files from archive 'AQ2:DiskSalv/Executive.lzx':
(      0 /      0) executive_v2.10/empty/ (      0 /      0) executive
*** LZX: Error writing to disk; disk full?
(      0 /      0) executive_v2.10/c/empty2/empty3/ (      0 /      0)
*** LZX: Error writing to disk; disk full?
** 30 file(s) extracted, all files OK
```

```
Operation successful
```

```
LhA Evaluation V1.51 - Copyright (c) 1992 by Stefan Boberg.
All rights reserved. Not for commercial use.
```

```
Creating new archive 'Executive.lha':
```

---

```

Frozen: ( 52.4%)    3604 => 1713 : executive_v2.10/.Product-Info
Archiving empty directory `executive_v2.10/c/empty2/empty3'
Frozen: (  6.0%)    3576 => 3360 : executive_v2.10/data/DashboardPrefs.1
Stored: (  0.0%)   38661 => 38661 : executive_v2.10/data/Register.lzx
Archiving empty directory `executive_v2.10/empty'
30 files added, all files OK.

```

Operation successful.

As can be seen above, the empty directories I added ("c/empty2/empty3" & "empty") are still archived, so the message does not seem to affect anything. For clarity, the disk was not full.

Since having access to the AmiNET, I have found an additional bug, and that is AmigaDOS's and LZX's inability to handle tildes (~) in filenames. I found this when copying files from the PC's here at University to my disks to take back to my trusty Amiga, when if a filename is greater than 8 characters long (suffix not included), it is truncated: so for instance an archive called "longarcname.lha" would be renamed "longar~1.lha" when saved to disk. When I tried to use these in LhA2LZX, LZX would not do anything with them, but gave an output of Operation successful. I have tried getting round this by detecting a tilde in a filename and getting the user to input a new filename, but AmigaDOS treats tilde's as something to do with negation - it is used in pattern-matching. If you want to get a list of all the files in a directory, you would type

```
list #?
```

If you wanted a list of all the files except for the icon files, you would type

```
list ~(#?.info)
```

which would list all the files apart from those ending in info (icon files). So it seems if an archive has a tilde in it's name, AmigaDOS and LZX try to operate on it in a pattern-matching context.

Using the Type command was the closest I got (e.g.

```
Type FROM arc~2.lzx TO arc2.lzx
```

but the command only copied part of the file. I tried also copy, join and rename, but these didn't change the filename either. The only way to change the filename via the CLI would be to use the apostrophe (') before the tilde e.g. if your archive was called my~arc.lzx:

```
Rename my'~arc.lzx TO myarc.lzx
```

I don't think it's possible to insert characters like this via a script :^(

You can of course also rename by the easy way - select the archive in it's window and go to "Rename..." from the Icons menu :^)

Thanks to

Frank Bunton  
for the help here.

This AmigaGuide has been tested only on WB3.1, so it is possible not all of the keywords will work under 3.0 or below systems. If anyone can read this on any other WB version, please contact me (you can load this into a text viewer and read what all the missing words, if any, are. If all is not well, I will change it.

Bug reports and suggestions of improvement are welcome.

Feel free to try out the first versions of LhA2LZX (i.e. 1.x), but only use copies of your archives as they have bugs in - hence you can see where I improved it... :^)

## 1.9 Program History & Development

### Program History

Versions 1.1, 1.2,

2.0

-

2.7b

are included in this distribution so you can see how it developed. Files are commented from 2.0+.

Version No.

~~~~~

1.0 Initial incarnation

1.1 Added filesize difference to show which archive was smallest

1.2 Fixed bug of 'l2z' not being deleted if the original archive was smaller. The following extract is from the original docs:

```
"- DO NOT QUIT WHEN DE-ARCHIVING otherwise ONLY THE FILES CURRENTLY
DE-ARCHIVED WILL BE PLACED IN THE NEW ARCHIVE. The new archive will not
contain all the original files and will overwrite the ORIGINAL archive.
```

So DON'T DO IT!

```
This happens because it is an AmigaDOS
script
```

```
file, and will perform
```

```
the next action immediately after completing the previous one. Thus if
it is de-archiving and it is interrupted by a CTRL-C, it will perform
the next action of archiving everything it has (so far) de-archived."
```

I found that if I aborted either LhA or LZX whilst they were de-/archiving then some of the original files were lost and, due to the nature of LhA2LZX's workings, were irretrievable unless a backup was kept. LhA2LZX was such that after it had de-archived and archived, the new file would replace the old, so files would be lost if this process was interrupted - some of the original files would not be archived and then the new would replace the old. Not a good thing. This was fixed in

v2.1

.

2.0 Added the extra option to convert to the other filetype. This was done for 2 main reasons. Firstly in case the user merely wished to convert between the filetypes, and secondly so I could convert my .lzx files to .lha, allowing me to enter them interactively via DOpus5 and add to them - my version of LZXDir 26.6.95 v1.1 (modified by Stone-D) doesn't allow me to add to .lzx files interactively.

#### Note

Directory Opus is a filemanager which allows you to quickly and easily perform actions on files. LhADir & LZXDir are two add-ons ("modules") which allow you to enter an .lha/.lzx file as if it were a normal directory and perform actions on the files within, e.g. viewing a picture, as well as deleting and adding files to the archive as if you were merely transferring files from one directory to another. If I want to add a file to an .lzx archive however I have to convert it to an .lha archive due to the version I have. If anyone has an updated version of LZXDir, please feel free to send it to me.

This version of LhA2LZX also checked if the file was already of the type that the user wished to convert to.

- 2.1 Added routines to check if the FULL path of the archive was chosen from the file requester by checking the size of the path name. Also did the same to check that a file had been chosen. Altered the "Convert to. . ?" routine to automatically check the filetype as this made things more logical (previously you could try to convert an .lha archive to an .lha archive . . .), and also saved me from trying to remember what type I'd just converted the file to. The routine allowing the checking of the filetype was already being used in LhA2LZX, so it was really just a cut-&-paste job once I'd realised it.

#### \* MAJOR BUG FIX \*

Stumbled across the means to add error detection if something went wrong in the de-/archiving process! BIG thanks to Peter Bagnato and his

#### DOSMan

creation - an AmigaGuide detailing the usage of AmigaDOS commands, with examples - which is a boon to those of us that haven't got an AmigaDOS reference manual. Fixing this bug meant that if anything went wrong when de-/archiving (such as a user abort or a corrupt file), then LhA2LZX would stop immediately and so all archives, the new and the original, would remain intact. And all by adding only 6 lines of code . . . I could now sleep at night.

- 2.2 Deleted all the 'deletes' at the start of the file as I decided that they were unnecessary. If LhA2LZX had been quitted and then re-started, the env: files would have already been deleted when it was quit the first time. If it was being started for the first time, chances are the files wouldn't be there anyway as env: is a temporary directory in RAM: and therefore erased with each reset. If the user chose to convert more archives (the final requester of the program), then the env: files would be overwritten with the new variables. This also allowed me to use the directory of the last archive chosen as the default path for the next archive - something I had wanted to do from the start - as it saves on searching through all of your directories if you have a drawer with several archives in which you wish to convert.

I also changed the new archive copying routine to copy the new archive to the selected destination BEFORE deleting the old in case of mishaps, but then altered it back for two reasons - there may not be enough space in the destination, especially if it is a floppy, to contain both the archives (new & old) before the old is deleted, and also it would now make little difference as this part of the script would never

be reached if something DID go wrong when it was de-/archiving, now that the error detection was in place.

The "Convert to . . ?" requester was changed to show the filename in the title bar. The output messages detailing which archive is smaller were put into requesters to lessen the output in the CLI window - but I had to sacrifice the ANSI colouring to do this. Now the only CLI output to screen is the coloured closing message, which is displayed whilst all the temporary files created are deleted.

Amount of free memory is checked if the 'l2z' is selected in RAM:, to make sure you won't run out, and a requester pops up advising you if this might happen. I would also like to find the amount of free space of the destination directory (in case RAM: isn't chosen) to see if there is enough, but I haven't yet found a way to do this using the AmigaDOS files in the Workbench distribution.

2.3 BaseName now is no longer used, as a function of List is used instead, to give the filename without a suffix and the suffix on its own. Deleted a couple of deletions of environment variables as they were overwritten on the following line, and so were unnecessary. Added a small routine just to check that the archive name does have an lzx/lha/lzh suffix (lzh is de-archived by LhA). Added detection to check firstly if the alternate path for 'l2z' exists and secondly that it is the full pathname.

2.4 LZH archives now processed fully as if they were normal LhA archives. Found a much nicer line to clear the screen for the final Thankyou message (from an old Amiga Power coverdisk!). Extra options added to the LhA/LZX command line to give faster progress display and suffixing. Added extra ANSI line "\*e[0 p" which supposedly turns off the cursor to make text output twice as fast, so LZX etc. can now work faster. Empty directories now dealt with and de-/archived as usual, although there seems to be a

bug  
in LZX here - luckily this seems to in no way affect  
the running of LhA2LZX.

Also added a similar routine to that of the default path of the archive when using LhA2LZX for multiple archives (see

v2.2  
) to have a default  
path for where 'l2z' is stored, in case the user doesn't have much RAM: (and so won't want to go searching through drawers each time they want to have 'l2z' somewhere on their hard-drive).

2.5 Added 2 small routines just to check that both the path and the archive exist when selected, in case the user decides to type in either. Also improved the "Convert to . . ?" requester to show exactly what filetype you are converting and what you can convert it to. This makes things much easier to see (in terms of options), and also shows off the ability of filetype detection more.

Also fixed a small bug in the Insufficient Memory area to do with quotes and volume names with spaces in, such as RamDisk:. Filenames or volume names with spaces need to be surrounded by quotation marks, like this:

"Ram Disk:"

so they can be accessed -if you just type Ram Disk: in a Shell, AmigaDOS will try to run the command Ram. Now the path name of the archive can have spaces in.

## 2.6 First official release (on my web page)

As the keyfile for LZX is now freely available on the AmiNET, extra features of LZX are enabled, such as de-archiving of LhA/LZX files, with a 25-30% speed increase. Also is the option of faster archiving at the cost of 32k of RAM. I have included both of these here to increase the overall speed of LhA2LZX. I tried here to fix the tilde

bug

, but

without much success. Expect better handling in the next release!

### 2.6a First AmiNET upload

All I did here was shorten the length of the program by renaming the variables with the prefix "nl2z", which is unique enough for me to use list to generate a list of the variables at the end and use this in a script created by LhA2LZX to delete them all. I got this idea from reading the startup-sequence (it uses the same principle but runs monitor drivers). I also changed the command that clears the output screen at the end to make sure it will work on KS2.0.

## 2.7 This version now contains filename error trapping, and fixes the tilde bug for the moment. If a tilde is detected in the filename, an error message is produced and the program quits. I am currently trying to think of a way round this without quitting the program or have the user typing an apostrophe before the tilde themselves, but I really don't know of a way to get AmigaDOS to scan filenames and insert specified characters before other characters :^( , and I don't really want to use AREXX or an actual program to do this as LhA2LZX is not just a file conversion utility, it is an exercise in AmigaDOS, showing its potential and all that it can do.

LhA2LZX now skips to the end and deletes all variables currently made if LhA or LZX cannot be found, instead of just quitting.

As the registered version of LZX is now needed, a quick check is now performed to see if the lzx.keyfile exists, as otherwise LZX won't work.

### 2.7a It seems that the change to de-archiving with LZX instead of LhA has revealed a bug in the 68000 version of LZX, which seems to ignore some of the command line options for de-archiving, or at least not process them correctly as the 68020 version does. The filename of the archive is given to LZX, but the '000 version adds the suffix ".lzx" to the file even if it is specifically told not to. This means it cannot find the file (because it is looking for "myarc.lha.lzx" instead of "myarc.lha"), and so it returns a Return Code of 10. If there is a user-abort, a Return Code of 20 occurs (Return Codes are values passed back from programs to indicate the result of an operation - such as successful, fail, user - abort and so on), which is what LhA2LZX checks for, as I found previously nasty things happened if a user aborted the program (ie the archive was deleted). I tried to also check for this RC of 10, but found that when checking a Return Code only one could be checked. Knowing already that a 0 is returned if the operation is successful, I altered the Return Code checking line to inversely check - ie check that the operation was



not successful, and act accordingly. This would mean that if anything went wrong (ie something that had not happened during my tests), LhA2LZX should compensate and Do The Right Thing, even if I had not anticipated the problem or even known about it. Now LhA2LZX pops up an error message requester telling you of the problem, and then skips to the "More...?" requester. Your archive should remain intact, although not converted, and will not be deleted as it would previously have been. If you are using an '020 or higher, you will probably never see this requester.

I also changed part of the LZX command line as a result of this (the LZX command line refers to the line of code in LhA2LZX in which LZX is started and various commands and options are passed to it - for instance what the archive name is, whether to archive everything in sub-directories and so on). I changed the -X option to -X0, which would not automatically append the suffix ".lzx" to the archive filename - this seems to change nothing if using the '020 version (as long as you don't delete the suffix in the requester, but you wouldn't do this anyway as you can just click on the filename), but if using the '000 version it will return an error. Strangely, the error returned is "Bad LZX archive header" - not surprising because it is operating on an LhA archive! The '020 version handles LhA files the same as LZX files. It seems that the '000 version in the LZX distribution doesn't have the registered features...

I had a cunning idea to use the Version command to determine which version of LZX is being used, but the only version information given by LZX is the version number. There are other ways of finding which version (like checking the size of the LZX executable using List or something), and changing the command line accordingly (ie using LhA to de-archive on an '000), but then how many people with '000s nowadays would be using this program? Thus I have decided that LhA2LZX now requires an '020 or faster CPU, as this is what most have nowadays. If you're still using an '000, write to me (snail-mail or e-mail) and I'll send you a version for it.

Another small change was just putting the path of LhA and LZX in speech marks, just in case your path has a space in (like if you keep them in "Workbench 3.0" or something).

I shall also note that LhA2LZX supports filecomments and filedates and so on of the files in the original archive - this means all comments etc will stay intact when converted between the two archive types. I am told not all the programs similar to LhA2LZX do this (although that may change after the other authors' read this...).

2.7b Not much change here in terms of code, but a bit in ideas. The suggestion came from Mosquito, and it seemed like a good idea to implement. The file comment, creation date and time of the new archive are replaced by the old - so the only noticeable difference between your archives is the file type and the file size (and the name slightly, obviously!). Thus all file comments of the archives remain intact, and the old date used makes it look as if it was the new archive you downloaded/created in the first place! This was all done by using features of the List command - showing just how versatile it is!

Another tiny detail was changed - after converting to the other archive type, I used the suffix .(lha|lzx) (lha or lzx) to copy the new archive

---

over and get its size:I used this format because you were converting to the other file type, which could have been lha->lzx or lzx->lha. Since writing that line I had added another variable somewhere along the way which would give the other file type suffix (\$nsuf), so I altered the line to give the other suffix, instead of either because I could now determine which you had converted it to.

## 1.10 For the Future

For the Future

- Even safer archive handling by keeping backups.
- New routines to copy multiple archives from floppy disk to hard drive, and copy the individual archives back after processing.
- New routines to check new archive sizes to see whether they will fit nicely back on floppy.
- Greater error trapping and handling, resulting in alternative directories being chosen to store the new archive
- Better handling of the tilde bug

What I would like...

...if anyone can help:

-A small program to get the space available on a disk and output only this (preferably using it's Filesystem, in case it's using something like PFS and the diskspare.device...). This will allow me to tell the user information such as if the new archive will definitely fit on their floppy, how much space is left and so on)

-Anything else you think might be useful to me... #&^)

## 1.11 Legal Stuff - Disclaimer

Legal Stuff

Standard disclaimer:

THERE IS NO WARRANTY FOR THE SOFTWARE TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHERE OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE SOFTWARE "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOFTWARE IS WITH YOU. SHOULD THE SOFTWARE PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY REDISTRIBUTE THE SOFTWARE

AS PERMITTED BELOW, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE SOFTWARE (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE SOFTWARE TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

LZX is © 1995 Data Compression Technologies  
LhA is © 1992 Stefan Boberg

All executables used in LhA2LZX, except  
LhA  
&  
LZX  
, are ©1986-1992 Commodore  
Amiga Inc. All Rights Reserved.

LhA2LZX is © 1998 Richard Burke

## 1.12 What I Used

### What I Used

BED - Black's Editor, to write LhA2LZX and also this AmigaGuide doc. It is not your average PD editor, so get it soon! It allows you to edit text/binary files, has a user-configurable menu bar, has a multitude of AREXX macros (and you can add your own) and can do just about anything you require - justification, clipboard support, encryption & viewing individual AmigaGuide pages as you write them are just a few of the options!

DiskSqueeze -an AmigaDOS  
script  
file which uses LZX to compress entire disks  
and so giving better compression than DMS! It also allows a filelist to be added to the archive, so that you know exactly which files are on the disk. DiskSqueeze helped me to understand how to structure a script file (although originally it was by messing around with  
DMSMaker  
that I  
learnt how to write scripts) and gave me a few ideas to improve LhA2LZX (for instance using the Which command, checking memory size and most notably about the use of  
Environment Variables  
).

DOSMan - An AmigaDOS reference manual (in AmigaGuide and MUI format) which details the usage of commands up to and including the new WB 3.1 ones, and gives examples of how to use them. Invaluable if you don't have one, and essential to find out what all the  
CLI  
commands do and how to use  
them.

And, of course ...

LhA  
Evaluation V1.51

LZX  
1.21 (Evaluation) Archive/Extract utility - 68020/68030 Version.

## 1.13 Thanks to . . .

"LhA2LZX has been brought to you today by ..."

Thanks to

Marco Negri for  
Black's Editor  
v1.02

Dirk Vael for  
DiskSqueeze  
v1.22

Peter Bagnato for  
DOSMan  
v1.21

Stefan Boberg for  
LhA  
v1.51

Data Compression for  
LZX  
v1.21

Technologies

Jotes of Fraxion for his Ansi.Manual v0.9, allowing text clearing and  
coloured text!

Frank Bunton for his help with the  
tilde bug

<http://www.acay.com.au/~bunton/>

Bill Duxbury for his help in finding the '000 LZX  
bug  
And finally...

to Sarah McLachlan, for singing to me for countless hours as I toiled  
away at my keyboard. Not personally, mind... (her piano wouldn't fit  
in my bedroom)

## 1.14 frank

Wahey! Told you I'd put it in. The only way to read this is either to  
double-click on your name in the "Thanks to..." node, or to read this .guide  
in PPMore or something. I wonder if I'm the first to ever do this? You can  
click on the Browse buttons all you want, but you'll never get to this page!

## 1.15 Contacting the Author

Contacting the Author

If you have any suggestions, comments or bug reports, you can contact me at the following addresses:

e-mail: R.Burke-97@student.lboro.ac.uk

Snail-mail: c/o 37 Woodside Gardens  
RAVENSHEAD  
NOTTINGHAM  
NG15 9GF  
UK

E-mail is preferred if possible (but don't expect a reply during the Summer months ... :^) ... and that means June - September for all you Southern - Hemispherists...).

If writing by snail-mail, please include a stamped addressed envelope.

LhA2LZX is GIFTWARE. That means if you use it, send me a little something. It doesn't matter if it's a nice picture by e-mail, a link to a web site I might like, or even a 20p bar of chocolate (mint's my favourite...). I don't care, as long as I know you are appreciating my small gift to you :^)

You can download the latest version from my LhA2LZX web page:  
<http://www-student.lboro.ac.uk/~corbl/index.html>

## 1.16 Also by the Author

Also by the Author

All the programs below are AmigaDOS  
script  
files created purely to show what  
you can achieve, using the software you already have, to make your life easier.  
Most of them are simply "front-end" programs to  
CLI  
executables- so you just  
click requesters instead of typing out CLI commands like in LhA2LZX.

AllowBad.req

Do you have floppies with hard errors on? Do you find that you simply can't format them or use them properly? Then wipe away your tears, for AllowBad.req is here! A simple requester interface to Mikolaj Calusinski's AllowBad means you too can now live with the peace of mind that never again will you have a Bad Floppy! ®

DMSMaker

A fab little AmigaDOS script file allowing you to use all the standard functions of the DiskMasher System - including read, write, view, test, repack, and even encrypt your DMS files with an invisible password!

---

**DSQView**

A small file which finds the filelist in a DiskSqueeze archive and displays it.

**FillDisk**

Helps fill disks to their maximum, using Torbjörn Hultén's "Fill v2.0." Choose your files by requester, and Fill calculates which combination of files to use to fill the disk to its maximum. Useful for storage.

**HappySearch**

I once had a virus which adds itself to every file you open, and is recognisable by finding "Happy New Year 96!" in your files. HappySearch will search specified files to see if you have the virus.

**LhAMaker**

This creates LhA archives from specified input files, via requesters.

**UnLZXer**

This de-archives LZX archives to a chosen destination.

## 1.17 Archiving

### Archiving

Archiving means to backup files. In the case of LhA & LZX, all the files you wish to backup -store a copy somewhere in case you need them in the future to, for example, replace lost data- are merged into one archive file and compressed, so that in almost all cases the final archive file takes up less space than the files which were archived did.

See also

De-archiving

NB "De-/archiving" in this document refers to when LhA2LZX de- ↔ archives and

then archives an archive.

## 1.18 De-archiving

### De-archiving

De-archiving means to restore files from a backup archive. Files are decompressed and separated back to individual files.

See also

Archiving

## 1.19 The Command Line Interface

### CLI

The Command Line Interface allows the user to communicate to the computer

---

via the keyboard. The Shell is the Amiga's CLI.

## 1.20 LhA & LHZ

LhA is a utility which allows you to compress and archive files. It was a very popular program for backup and distribution purposes (i.e. the AmiNET) but is now being surpassed in terms of decompression times and archive size by

LZX

.

LhA archives can be compressed further. If you archive all your files into an LhA archive with no compression, then archive this with LhA with compression, you can often save quite a lot more space, although you will have to de-archive it twice. I have not included this facility in LhA2LZX, because the same does not work with LZX, and it would have no real indication how many times the archive had been re-archived, and may get too complex to be useful. But it's nice to know anyway :^)

LHZ is similar to LhA and files archived with this are de-archived by LhA.

## 1.21 LZX

LZX is a utility which allows you to compress and archive files with often a great degree of compression. It also decompresses data faster than any other popular Amiga program. It gives greater compression because it uses the knowledge of previous files in the archive, using an ability to transparently look for similar compressible data across multiple files before adding them to an archive.

See also

LhA/LZH

## 1.22 RequestChoice

RequestChoice brings up a requester in which the user is prompted to make a choice. The title-, body- and button-text is configurable so that it may be used in any situation. The value of the button pressed by the user is then returned so that the relevant process is chosen.

See also

RequestFile

---

## 1.23 RequestFile

RequestFile brings up a filerequester in which the user is prompted to choose a file. The title- and button-text is configurable so that it may be used in any situation. The chosen file name is then returned so that it may be used later.

See also

RequestChoice

## 1.24 Index

Index

- Read LhA2LZX
- AmigaDOS Script files
  - Archiving
  - Arguments
  - CLI
  - De-archiving
- Environment Variables
  - LhA/LZH
  - LZX
  - Problems
- RequestChoice
- RequestFile
- Requirements
  - Usage
- Program History -
  - v1.0
  - v1.1
  - v1.2
  - v2.0
  - v2.1
  - v2.2

---



v2.3

v2.4

v2.5

v2.6

v2.6a

v2.7

v2.7a

v2.7b

Contacting the Author