

UnSquish

Copyright © 1996-1998 Stephen Williams

COLLABORATORS

	<i>TITLE :</i> UnSquish		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		June 25, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	UnSquish	1
1.1	main	1
1.2	UnSquish 2.1 Introduction	1
1.3	UnSquish 2.1 Usage from Shell	2
1.4	UnSquish 2.1 Usage from Workbench	4
1.5	UnSquish 2.1 Usage from Workbench	5
1.6	UnSquish 2.1 Usage from Workbench	5
1.7	UnSquish 2.1 Usage from Workbench	5
1.8	UnSquish 2.1 Usage from Workbench	6
1.9	UnSquish 2.1 Usage from Workbench	6
1.10	UnSquish 2.1 Usage from Workbench	7
1.11	UnSquish 2.1 Usage from Workbench	7
1.12	UnSquish 2.1 Usage from Workbench	7
1.13	UnSquish 2.1 Usage from Workbench	8
1.14	UnSquish 2.1 Tooltypes	8
1.15	UnSquish 2.1 Limitations	9
1.16	UnSquish 2.1 Acknowledgements	9
1.17	UnSquish 2.1 Program History	10
1.18	UnSquish 2.1 Information	11
1.19	UnSquish 2.1 Legal Matters	12
1.20	UnSquish 2.1's Author	13

Chapter 1

UnSquish

1.1 main

UnSquish 2.1

by
Stephen Williams

Release date: August 17, 1998

Introduction

Shell Usage

Workbench Usage

Tooltypes

Limitations

Acknowledgements

Program History

About Compressors

Legal Matters

1.2 UnSquish 2.1 Introduction

What is UnSquish?

An interface to the XFD library, operable from Shell or Workbench. In plain English, this means that UnSquish is a virtually universal decompressor. It will unpack most packed files (but not archives).

How to install UnSquish

Drag the icon to wherever on hard- or floppy-disk you want to install it. You must also have xfdmaster.library version 37 or above installed in LIBS:. If you don't have a suitable version, one is supplied in the UnSquish/libs/ directory. Just copy it into LIBS:. xfdmaster.library is © of

Georg Hörmann

.

If you want to use UnSquish from

Workbench

, you will need

reqtools.library version 38 or above in LIBS: as well; UnSquish uses reqtools.library for various functions. I have supplied a recent version of reqtools.library in UnSquish/libs/. reqtools.library is © of

Nico François

and

Magnus Holmgren

.

UnSquish requires AmigaOS release 2.00 or newer (Exec version 36 or greater).

Unpacking XPK files with UnSquish

In order to unpack XPK files, you must have xpkmaster.library installed in LIBS:, and the relevant sublibraries installed in LIBS:compressors/. These files are not included; anybody using XPK files will have them already :-)

Next section:

Shell Usage

1.3 UnSquish 2.1 Usage from Shell

Previous section:

Introduction

How to use UnSquish from Shell

Format: UnSquish [source file] [destination file] [PASSWORD=<password>]

Template: UnSquish SOURCE,DESTINATION,PASS=PASSWORD/K

UnSquish will read the file specified by SOURCE into memory and attempt to decompress it. If decompression is successful, UnSquish will write the decompressed file out to disk with the name specified by DESTINATION. If DESTINATION was omitted, UnSquish will replace the original file. UnSquish will ask for confirmation before overwriting any file. Partial files written as a result of a failed save will not be deleted; this "feature" may be changed in a future release.

DESTINATION can be a directory instead of a file, in which case UnSquish will save the new file into this directory, with the original name.

Examples

To decompress file "foo":

```
UnSquish foo
```

To decompress file "foo", saving the new file with name "bar":

```
UnSquish foo bar
```

To decompress file "foo" in the C: directory, saving the new file as "foo" in the current directory:

```
UnSquish C:foo ""
```

or

```
UnSquish C:foo foo
```

To decompress file "foo" in the current directory, saving the new file in the RAM disk:

```
UnSquish foo RAM:
```

or

```
UnSquish foo RAM:foo
```

To decompress file "foo" in the current directory, saving the new file in the RAM disk with name "bar":

```
UnSquish foo RAM:bar
```

The PASSWORD parameter

If the compressed file has been encrypted (eg with PowerPacker, or XPK FEAL), a password must be supplied for UnSquish to be able to decompress the file. This is done by putting the password on the command line, prefixed with "PASSWORD=". The abbreviation "PASS=" may be used. Upper or lower case may be used for the prefix, though the password may be case sensitive, depending on the encrypter used.

If a password is specified for a non-encrypted file, it will have no effect.

You may not be too keen on the password being visible on the command line.

Workbench

usage of UnSquish circumvents this problem.

Examples

Decrypt the file "foo" in the current directory, with password "bar":

```
UnSquish pass=bar foo
```

or

```
UnSquish password=bar foo
```

or

```
UnSquish foo pass=bar
```

or

```
UnSquish foo pASswOrd=bar
```

etc etc... :-)

Starting UnSquish from the Shell with no command-line options will cause it to behave as if it has been started from

```
Workbench
.
```

Next section:

```
Workbench Usage
```

1.4 UnSquish 2.1 Usage from Workbench

Previous section:

```
Shell Usage
```

```
How to use UnSquish from Workbench
```

While UnSquish is a perfectly good

```
Shell
```

```
decompresser, its real purpose
```

is as a GUI interface to xfdmaster.library.

Run UnSquish by double-clicking its icon. A window will open, resembling the following wonderful piece of ASCII art:

```

[]=UnSquish 2.1=====[][]
||                               ||
||                               ||
    Filename
    Size
        ||
||                               ||
||                               ||
    Status bar
        ||
||                               ||
||                               ||
    Load
    Save
    About
    Quit
        ||
||                               ||
++-----++

```

Click on a window feature for information on that feature.

UnSquish also has a

```
menu
, and several
```


keyboard shortcuts
.

Next section:

Tooltypes

1.5 UnSquish 2.1 Usage from Workbench

The Filename display

This display contains the name of the file currently loaded
. If no file is loaded, the display will contain "No file loaded".

[Back to Workbench Usage](#)

1.6 UnSquish 2.1 Usage from Workbench

The Size display

This display contains the size in bytes of the file currently loaded
. If no file is loaded, the display will contain a zero.

When a file is first loaded, this display will contain the initial (ie compressed) size of the file. After decompression, this display will contain the decompressed size of the file.

[Back to Workbench Usage](#)

1.7 UnSquish 2.1 Usage from Workbench

The Status display

This display contains information about the current status of the program (eg "Ready", "Waiting for input"), to complement the information given by various requesters that appear during the normal course of operation.

Back to Workbench Usage

1.8 UnSquish 2.1 Usage from Workbench

The Load button

Pressing this gadget will open an ASL file selector, allowing the user to select a file to decompress. You may also select a file by dropping its icon into UnSquish's window.

Once a file has been selected, any existing file in memory will be purged, and the new one loaded. If it is compressed, UnSquish will show a requester asking you if you want to decompress it. Canceling this requester will unload the file; confirming it will decompress the file. Once the file has been decompressed, UnSquish will return control to you, allowing you to save the decompressed file, or select another one.

If the compressed file is encrypted, UnSquish will show a ReqTools string requester for you to enter the password. The password will not appear on the screen as you type it in.

You will be informed of any errors which occur during decompression.

WARNING: UnSquish will not ask for confirmation before purging any file in memory before loading a new one. It is up to you to make sure you've saved a decompressed file before loading another.

Back to Workbench Usage

1.9 UnSquish 2.1 Usage from Workbench

The Save button

Pressing this gadget will open an ASL file selector, allowing you to pick a filename to save the decompressed file to. The file will not be purged after a save, so you can make as many saves as you like.

You will be informed of any errors which occur during saves. A save error will not cause the file in memory to be purged; only selecting another file, or quitting the program, will cause this. Any partial file written as a result of a bad save will not be deleted; this "feature" may change in a future version.

If the original file had an icon, you will be given the option to save a copy of this icon with the new file.

If there is no file in memory to save, the Save button will be disabled.

Back to Workbench Usage

1.10 UnSquish 2.1 Usage from Workbench

The About button

Pressing this gadget will open the obligatory information requester.

Back to Workbench Usage

1.11 UnSquish 2.1 Usage from Workbench

The Quit button

Pressing this gadget will quit UnSquish. There is no "Are you sure" requester, so make sure you've saved any file in memory before you quit the program.

Pressing the close gadget on the window's title bar has the same effect as pressing the Quit gadget.

Back to Workbench Usage

1.12 UnSquish 2.1 Usage from Workbench

The Project menu

```
+-----+
| Load...  A O| · equivalent to pressing the
                Load
                gadget
| Save...   A S| · equivalent to pressing the
                Save
                gadget
+-----+
| About...  A ?| · equivalent to pressing the
                About
                gadget
```

```
+-----+
| Quit      A Q| · equivalent to pressing the
                Quit
                gadget
+-----+
```

Back to Workbench Usage

1.13 UnSquish 2.1 Usage from Workbench

Keyboard shortcuts

The following keyboard shortcuts are recognized:

```
RCOMMAND-O - equivalent to pressing the
                Load
                gadget
RCOMMAND-S - equivalent to pressing the
                Save
                gadget
RCOMMAND-? - equivalent to pressing the
                About
                gadget
RCOMMAND-Q - equivalent to pressing the
                Quit
                gadget
ESCAPE      - equivalent to pressing the
                Quit
                gadget
```

Back to Workbench Usage

1.14 UnSquish 2.1 Tooltypes

Previous section:
Workbench Usage
Tooltypes

UnSquish recognizes one Tooltype at present: TOPAZ.
UnSquish will attempt to adapt its GUI to the screen font, provided that the screen font is smaller than 20 points. If the screen font is larger than this, UnSquish will use the Topaz 8 point font instead. Setting the TOPAZ Tooltype instructs UnSquish not to try to adapt to the screen font whatever size it is, but to always use the Topaz-8 font.

Next section:

Limitations

1.15 UnSquish 2.1 Limitations

Previous section:

Tooltypes

Bugs and Limitations

UnSquish cannot be interrupted during a decompression. I know of no way of interrupting `xfdmaster.library` while it is decompressing. If I ever find a way, I will incorporate it into UnSquish.

UnSquish does not support the decompression of files encrypted with a numeric key, rather than a password. `xfdmaster.library` supports this feature; UnSquish does not because I do not have any of these encrypted files to test it with. If and when I obtain a suitable encrypter with which to test this feature, it will be supported.

Executables that have been packed by
absolute address packers
get

irrevocably altered by them. Whilst UnSquish will unpack the programs, they will not work in their decompressed state. Fortunately, not many people use this kind of packer any more. The most common Amiga packers (PowerPacker, Imploder, XPK) are perfectly safe, as far as I know. Georg Hörmann's program `xfddecrunch` is the best I know of for attempting to repair programs broken by absolute address packers.

People who have been using `xfdmaster.library` for some time will know that extra decrunch modules (called "slaves") are available for it. Whilst UnSquish does try to use these slaves, I have had no luck with them. This seems to be a problem with the slaves themselves rather than with UnSquish; `xfddecrunch` by Georg Hörmann gave the same errors. This should not present much of a problem, as all the common decrunchers are included in `xfdmaster.library`. Only a few obscure ones exist as slaves.

Next section:

Acknowledgements

1.16 UnSquish 2.1 Acknowledgements

Previous section:

Limitations

Acknowledgements

UnSquish uses decompression routines from xfdmaster.library, © of Georg Hörmann. Without Georg's superb coding, UnSquish would not have been possible.

reqtools.library is © of Nico François and Magnus Holmgren.

Thanks must go to my beta-testers:

Dave Hardenbrook
Dave Jones
Dave Manley (all these Daves... :-)
Ian Gledhill
Chris Underwood
Glen Watts

Without the help of these people, UnSquish would contain more bugs than an antmound.

Next section:

Program History

1.17 UnSquish 2.1 Program History

Previous section:
Acknowledgements
Program History

- Version 0.1 (21.8.96, beta release)
 - First release.

- Version 1.0 (25.9.96, first public release)
 - Small bug in window drawing code fixed.
 - Now requires xfdmaster v37+ (previously required v36+).

- Version 1.1β1 (11.10.96, aborted beta release)
 - UnSquish now saves files with the same permissions as the loaded file; ie if a crunched file had protection set to hprwed, the decrunched file will have the same permissions. Previously, all saved files had permissions set to rwed. The exception is that UnSquish will always clear the archive "a" bit. This is standard behaviour for a program which saves files.
 - If invoked on a pre-AmigaOS 2 machine, UnSquish now simply exits, instead of reporting an error. The error message had been sent to STDOUT; because the routines to write to STDOUT are quite large, this alteration removes about 2.5Kb from the size of UnSquish! I decided that keeping the program small was more important than writing this error.

Version 1.1β1 was never released. I was not satisfied with the operation of version 1.x of the program; the GUI was downright lame. UnSquish was put on hold, until...

- Version 2.0β3 (26.3.97, beta release)
 - Program totally rewritten from scratch.
-

- Development language changed from Pascal to C.
 - Separate Shell and Workbench functionality.
 - Proper Amiga GUI (menus, buttons, drag and drop).
 - Font-adaptive GUI.
 - Support for password-encrypted files.
 - Support for external XFD "slaves".
- Version 2.0 (26.6.97, public release)
- Public release of version 2.0B3.
- Version 2.1B1 (11.9.97, beta release)
- Substantial behind-the-scenes code cleanup.
 - Now responds to shift-clicked icons when started from Workbench.
 - Now opens GUI window if started from Shell with no arguments.
- Version 2.1B2 (internal release)
- Okay, so I haven't altered UnSquish for the best part of a year. Yup, I'm a slacker :-)
 - More code re-organization.
 - When busy, locks the window instead of disabling the gadgets (neater).
- Version 2.1B3 (10.7.98, beta release)
- Big code reorganization. Big enough to warrant a revision number bump to 2.5, if I were that way inclined.
 - The "about" requester is now synchronous. Never should have made it asynchronous in the first place. Stupid feature.
 - Now uses ASL file selectors instead of ReqTools ones.
 - Ditched the annoying "You did not select a file" requesters that popped up whenever you canceled a file selector.
- Version 2.1 (17.8.98, public release)
- Public release of version 2.1B3.
 - Minor bug fix - UnSquish wasn't setting protection bits properly when used from Shell.
 - Recompiled with xfdmaster v38 includes.

Next section:

About Compressors

1.18 UnSquish 2.1 Information

Previous section:
Program History
About Compressors

File packers are great. However, some are more efficient than others. As an example, take the old packer "Turbo Squeezer". Whilst it packs pretty well, newer packers (such as my favourite, Imploder) are more efficient, both in terms of file size and speed. However, Imploder has no option to decrunch files packed with old packers. This is where programs like UnSquish come in. Armed with UnSquish, you can decompress programs compressed with old packers, and recrunch them with Imploder (no, the authors of Imploder (Peter Struijk and Albert

Brouwer) are *not* paying me commission! :-)

Many old packers produce executables that break on newer versions of the Amiga's operating system. As long as the original file was not altered by the packer before packing, UnSquish will be able to restore these programs simply by unpacking them. You will then be free to repack them with a more modern packer if you wish.

Imploder does not support data files, only executables. For data, I recommend the XPK system. If you have xpkmaster.library installed, XFD (and hence UnSquish) will decompress files compressed with any of the XPK slaves that you also have installed.

Isn't it confusing that XPK and XFD look so similar :-)

Never use dirty "absolute address" packers. It is impossible to decrunch them properly. Programs like xfddecrunch (by Georg Hörmann) can *sort-of* get them decrunched and working. But it isn't easy; UnSquish doesn't even try.

Next sections: Legal Matters

1.19 UnSquish 2.1 Legal Matters

Previous section:
About Compressors
Legal Matters

xfdmaster.library is copyright by Georg Hörmann;
reqtools.library is copyright by Nico François and Magnus Holmgren.

UnSquish is copyright ©1996-1998 by Stephen Williams.

UnSquish is not in the public domain. The author, Stephen Williams, holds the copyright. UnSquish may not be altered in any way. No charge may be made for UnSquish's distribution, other than a reasonable charge for magnetic media. No profit may be made from UnSquish.

This version of UnSquish is a public freeware release. It may be distributed by FTP, on bulletin boards, by PD libraries, or on magazine cover (disks|CDs). The author is reasonably satisfied that it is bug-free and stable.

The author can not be held responsible for any loss, injury, damage or swan obsession caused as a direct result of using UnSquish.

Back to top

1.20 UnSquish 2.1's Author

How to contact the author

Send snail-mail to:

Stephen Williams,
9, Helmsley Way,
Spalding,
Lincolnshire,
PE12 6BG,
England.

Send email to one of the following addresses:

sw@nysa.u-net.com
sw@fysh.org

Web presence:

<http://www.nysa.u-net.com/>

Other software by Stephen Williams

Cookies v4.5 (Aminet util/misc/Cookies.lha)

A fun little fortune cookie program. It can output to a Shell window, or open a ReqTools requester. It uses its own database format, but comes with a utility to convert the common "%" database files into its own format. It has a fast search routine, hand optimized in assembler; searches are virtually instantaneous even on an unexpanded A1200! Cookies supports database files compressed with XPK and PowerPacker. AmigaOS release 1.3 or greater is required to run the program; some features require AmigaOS release 2.0 or greater.

Switch v1.2 (Aminet util/misc/Switch.lha)

A metric conversion utility that will convert between a wide variety of British and metric measures (for example, inches to centimetres, pounds to kilograms, fahrenheit to centigrade). You can print out the conversions you make. Requires AmigaOS release 2.0 or greater.

Zeller v2.0 (Aminet util/time/Zeller.lha)

A perpetual calendar, able to display any month for any year between AD1753 and AD2100. You may print out any month you like. Uses MUI for its user interface. Requires AmigaOS release 2.0 or greater, and preferably a reasonably fast processor and some fast RAM, as MUI programs can seem sluggish without these.

File v1.0 (Aminet util/misc/File.lha)

A file identification utility, like the Unix command of the same name. It uses FileID.library to identify files. It can be used from the Shell like Unix File, or from the Workbench, where it features an

appwindow. AmigaOS release 2.0 or greater is required.

NIPrefs-GT v1.0 (Aminet util/?/NIPrefs-GT.lha)

NIPrefs-GT is a replacement for the standard NewIcons V4 prefs editor. While the original uses ClassAct for its GUI, my version uses GadTools, and is therefore smaller and quicker. To use NIPrefs-GT, you will need NewIcons V4 installed (why would you want to use it otherwise ;-). You will also need AmigaOS release 2.04 or greater.

Exmoor 1.0 (Aminet text/show/Exmoor.lha)

An XPK wrapper for More, allowing More to be used to read XPK compressed text files. Once installed, you just type "Exmoor" wherever you used to type "More", and it all works transparently. Exmoor requires the XPK package to be installed (you can get it from Aminet), and needs AmigaOS release 2.04 or greater.

LandFill v1.0 (Aminet util/wb/LandFill.lha)

A Workbench appicon trashcan. No really special features; it's small, efficient and tries very hard to erase each file or directory you ask it to, unlike some appicon trashcans, which try once and then give up. LandFill needs AmigaOS release 2.04 or greater.

WBLaunch 1.0 (Aminet util/cli/WBLaunch.lha)

Uses Stefan Becker's wbstart.library to run applications from the Shell as if they had been run from Workbench. It is useful when using Shell-based development tools to test a program's Workbench behaviour. It may also be used in IconX scripts to start a program as if from Workbench, or simply used from the command line to launch a program which ignores its ToolTypes when started from the Shell. Requires wbstart.library (Aminet util/libs/WBStart.lha) and AmigaOS release 3.0 or greater.
