

UNd64

COLLABORATORS

	<i>TITLE :</i> UNd64		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		October 9, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1 UNd64	1
1.1 UNd64 Guide, yea yea yea...	1
1.2 UNd64 Guide, yea yea yea...	1
1.3 DISCLAIMER:	2
1.4 DESCRIPTION:	3
1.5 REQUIREMENTS:	3
1.6 Usage	3
1.7 <code>_FIRST_</code> there was UNd64...	4
1.8 <code>.d64/.x64</code> Support	5
1.9 WHAT ARE <code>.d64</code> DISK IMAGES?	5
1.10 <code>.X64</code> Notes...	5
1.11 <code>.d64</code> Notes...	6
1.12 <code>.d64/.x64</code> Output:	6
1.13 <code>!zipcoded RLE-Compressed Split Diskimage Support</code>	7
1.14 <code>.t64</code> Support	8
1.15 WHAT ARE <code>.t64</code> TAPE IMAGES?	8
1.16 <code>.t64</code> Output	8
1.17 <code>.t64</code> Notes...	9
1.18 <code>.p00</code> Support	9
1.19 <code>.lnx</code> Support	10
1.20 <code>.lbr</code> Support	11
1.21 <code>.sfx</code> support	11
1.22 Progress Messages	12
1.23 Compatibility	13
1.24 History:	14
1.25 TO DO...	18
1.26 Author/Updates...	18
1.27 And...The Quote of the Day:	19
1.28 The Amiga 600 Formerly From Hell	19
1.29 Thanks to...	19

Chapter 1

UNd64

1.1 UNd64 Guide, yea yea yea...

UNd64 v 38.18 (12.1.98)

© Copyright 1994-1998 Jess Sosnoski

This program has become emailware, if you would like to be considered a registered user, you have to send me email!

If you can't send me email, postcards or other forms of communication are perfectly acceptable.
(smoke signals, etc...)

Also, it will help me figure out what kind of user base this program has.

NOTE: I've seen some people referring to this program as version 0.xxx
It's not a 0.xxx version it's VERSION 38 REVISION 18.
Thank You :)

To The Docs....

1.2 UNd64 Guide, yea yea yea...

UNd64 v 38.18 (12.1.98)

© Copyright 1994-1998 Jess Sosnoski

Disclaimer

Description

Requirements

Usage

Progress Messages

.d64/.x64 Support

.z64/1!zipcoded Support

.t64 Support

.p00 Support

.lnx Support

.lbr Support

.sfx Support

Compatibility

History

TODO

Author/Updates.....

Quote of the day

FIRST there was UNd64..

Thanks to...

1.3 DISCLAIMER:

DISCLAIMER:

Use this program at your own risk!

I assume no responsibility or liability for problem(s) and/or damage(s) that occur by the use, modification, and/or existence of this product, and/or its parts in any form.

(I.E. If it makes you spill beer on your computer, ugh..the thought of it)

This program is copyrighted by Jess Sosnoski and is available as e-mailware.

It may be distributed only under the following conditions:

- 1.) Program and Documentation is not modified in any way.
- 2.) Program and Documentation must be distributed together.
- 3.) This package may not be used for commercial purposes.
Without my permission.
- 4.) I give permission to release this on CD-ROM.
Especially Aminet and Fred Fish.
- 5.) This may be released on coverdiscs, however, you must contact me first to be sure you are using the latest version.
I'd also like a free copy of the magazine too...

/* I like the idea of being immortalized in a piece of laser-etched aluminum encased in nice non bio-degradable plastic. */

1.4 DESCRIPTION:

DESCRIPTION

This program is designed to augment c-64 emulators and/or real c-64s by giving you access to the files stored in various types of archives from different sources.

The currently supported archives are:

- .d64 disk images, 35-track variety; with or without extra error info.
- .t64 tapeimages
- .p00 program images from PeeCee C64 emulators
- .x64 Unix c-64 emulator diskimages
- .lnx C-64 LYNX disk archives (ALL versions, both punter & xmodem types)
- .lbr C-64 Library archive files
- .sfx C-64 SelF-eXtracting compressed archives
- !zipcode c-64 zipcoded RLE-Compressed split disk images.

You can find c-64 software galore around the net these days...just visit your local search engine, and look for the supported archive types, or even "64 software".

1.5 REQUIREMENTS:

REQUIREMENTS

An Amiga with Kickstart 2.x+

Enough ram to hold the file you want to process, some supported archive types, and a storage device that will hold them--and possibly the output files.

1.6 Usage

USAGE

From the Shell/CLI (only):

```
und64 [command]{option} [filename] {path}
```

If run from workbench, it yells at you.

If run without parameters, usage information will be displayed.

At any point during program execution, pressing CTRL-C will abort.

Commands:

- l List file(s) in archive.
- x Extract file(s) from archive.
- t Test file(s) in archive.

Options:

d Handle DELETED files (.d/x64 ONLY!)

* WARNING * Resurrecting DELETED files is risky business, if a new file is partially written over part of a DEL'd file, you may also get the end of another file as a surprise bonus. Or maybe

you'll just get a part of an already complete file.

a Handle ALL files. (.d/x64 ONLY!)

This will allow extracting of splat files, or in other words, incorrectly closed files. Same warning as DEL files. Splat files have a filetype prefix of *.

n No Filename Conversion. This leaves the filenames alone, however, they might look pretty weird in the Amiga's ASCII. Illegal characters are still replaced during extraction.

NOTE: Filenames are converted to AmigaASCII, and any characters <32 or >127 are converted to printable characters. Characters /"?:#*, are converted to -

Filename:

Name of the file including the extension or prefix (if exists).

Path:

The optional output path, if you don't want to extract to the same directory you have your archives in.

If the output path includes a directory, you must put a slash after it. i.e. RAM:junk/ otherwise (as with the above example, but with no trailing slash), it would extract to RAM:

* NOTE * the path may not exceed 255 characters or you may get unpredictable results. (I'd say 255 characters is a pretty fair limit, given the fact that nobody out there should want to bury c64 files that deep in directories)

Example:

```
und64 la FooBar.d64
```

or

```
und64 la !!blah
```

NOTE The program autodetects the filetype of the archive, and acts accordingly.

1.7 FIRST there was UNd64...

And it still IS UNd64!!!

There's another program out there which DARES to imply that it sends my program into obsolescence with this line, and considering my program (IMHO) is FAR more robust and idiot-proof (not to mention speed), I'd avoid this other program.

COMPARE AND SAVE !!!

BENCHMARKS:

Extracting from COMMODOR.D64 with output redirected to >NIL:
UNd64 6 seconds
Brand-X 35 seconds

I'll give the other program this: it very nicely "simulates" a 1541 drive, especially speedwise, and the way it reacts to circular links (I pity anyone with a SCSI-II and '060 equipped A4000T with a big hard drive. That other program should successfully fill that son of a b* up real damn quick.)

Oh, yea...my program's smaller too.

1.8 .d64/.x64 Support

UNd64 v 38.18 (12.1.98)
© Copyright 1994-1998 Jess Sosnoski

What's a .d64 diskimage?

Output

.d64 Notes

.X64 Notes

1.9 WHAT ARE .d64 DISK IMAGES?

WHAT ARE .d64 DISK IMAGES?

.d64 files are the sector-by-sector contents of a Commodore 1541 formatted disk.

These files are commonly used by PeeCee C-64 emulators to store C-64 files, and to xfer 1541 disks into a convenient method of storage for use by the emulators. (I believe the emulator is called C64S)

Unfortunately, few of our Amiga C-64 emulators can take advantage of this.

There also exists an archive on Aminet that allows one to create .d64 archives on a 5.25" drive equipped Amiga. The archive also contains a program for de-arc'ing files from .d64 archives.

(I believe it's in misc/emu/1541.lha)

WHAT ARE .X64 DISK IMAGES?

.X64 files are the diskimage format used by X64, a Unix C-64 emulator. They cover 1541, 1570, 1571, and 1581 formats.

1.10 .X64 Notes...

.X64 Notes

Basically a 1541/1570 .x64 diskimage is the same as a standard .d64 diskimage with a 64-byte header in front describing what type of image it is. For all intents and purposes, UNd64 just reads this header to make sure it's reading the right type, and then treats it as a .d64 file. Therefore.....for more info, check the .d64 section...

NOTE: UNd64 only supports 1541/1570 35-track diskimages.

.x64 files come in a variety of formats, handling 1571, 1581, and 40-track 1541 images. Maybe someday, I can add support for these as well.

1.11 .d64 Notes...

.d64 NOTES

There are several types of .d64 diskimages, they are as follows:
35-track standard, 35-track w/ appended error information (supported)
40 & 42 track standard, and 40 & 42 track w/ appended error info.
(not YET supported)

You might have noticed a couple of .d64's that only have one or two files on it, and that those files don't seem to do anything. Well, if this is the case, it is likely that the program(s) you extracted are the boot code for a trackloader-type program. This means that the information for the program is stored on the disk in another way besides a 1541-dos-compliant style. Basically, this means, you can't do much with them.

If you need to use them that badly, put the .d64 image back onto a 1541 disk, then use the 1541 thru A64's interface. There's a program to transfer .d64's to a 1541 drive on aminet called Easy1541.lha

There are also several other related utilities for doing things with .d64's on aminet...just look under misc/emu!

Also, don't try to extract d/x64'd GEOS disks either, they use a different method of file storage, and und64 will report file size errors. I can't guarantee the integrity of the data extracted from these diskimages.

1.12 .d64/.x64 Output:

Program Output:

UNd64 38.18 (12.1.98)
© Copyright 1994-97 Jess Sosnoski

Disk Name:old games 00 2a
 Approx

Blks	Bytes	Type	Filename
2	508	*PRG	truck and field
71	18034	*DEL<	track
97	24638	*PRG<	field
33	8382	PRG	spacedinvaders
33	8382	PRG	knight driver
33	8382	PRG	race
33	8382	PRG	wizzard of war
33	8382	PRG	kickinthehead
1	254	PRG	amigaforever!!
33	8382	PRG	bewareilive
66	16764	PRG	beamryder
3	762	*DEL	attack
55	13970	*DEL	.attack
4	1016	*DEL	kl
33	8382	PRG	ratrace
117	29718	PRG	jumping jacks j2
0	0	*DEL	froggee ii

Output Description

Blocks: number of disk blocks it took up on the disk.
A disk block=256 bytes, 254 used for data.

Approx Bytes: 254*Blocks. This is to give an approximate filesize
Actual filesize should be smaller than or equal to this.
NOTE: If someone messed with the dir structure and manually changed the number of blocks, this could give false information.

Type: This will be DEL, PRG, SEQ, or USR. This is for informational purposes only, and does not affect outputted files. For those of you out there who are not Commodore users, PRG files are typically programs/executables. SEQ files are typically text files. USR can be about anything. DEL files are deleted, you can list or attempt to extract them when using the d option. (DEL files are always marked as splat files) There are also REL(ative) files, which are not supported. However, the program will list them. (and ONLY list them). REL files are 3-d type files with all kinds of side sectors and records and junk--so I chose not to deal with them. (as if they could be properly stored on a filesystem which does not support them!) Splat files are prefixed with a *, and locked files are postfixed with a <. Splat files are improperly closed files or deleted files, and on a Commodore disk are read protected. Locked files on a Commodore disk are write/delete protected. These file attributes will not affect the output files either.

1.13 1!zipcoded RLE-Compressed Split Diskimage Support

I finally had time to add support for this type of archive, which seems to be increasing in abundance around the 'net.

Zipcoded files have nothing to do with .zip compression on PeeCeos or Un*x.
 (although you can find zipcoded files that have been .zipped....)
 But...they are: split compressed diskimages.

There are 4 files total in any zip archive, prefixed: 1! 2! 3! 4!
 When feeding und64 a zipcode file, you need only specify the first file.

Sometimes they have the extension .z64

Rules for valid zipcode archives:

All four files must exist and have the following properties:

Archive	Minsize	Maxsize
1!	508	43348
2!	506	43346
3!	518	44378
4!	527	45152

Und64 will load in the files, and internally convert them to a .d64, and then handle them as such.

Right now there is no option to explicitly convert zipcode->.d64 to disk.

The types of compression in zipcoded files are as follows:

1. Raw Storage (sector not compressed)
2. Static (filled with a single byte)
3. RLE-Compressed

1.14 .t64 Support

UNd64 v 38.18 (12.1.98)

© Copyright 1994-1998 Jess Sosnoski

What's a .t64 tapeimage?

Output

Notes

1.15 WHAT ARE .t64 TAPE IMAGES?

WHAT ARE .t64 TAPE IMAGES?

 .t64 files are a format developed by Miha Petemel for the PeeCee C-64 emulator C64s. Just another format that the emulator supports/uses to conveniently store C-64 files.

1.16 .t64 Output

UNd64 38.18 (12.1.98)

© Copyright 1994-97 Jess Sosnoski

Tape Name:imported files

Number of Entries: 30

Number Used : 1

Version : 1.0

Size	Type	Filename
37632	PRG	1a.C64

Output Description

Tape Name: Name that the creator of the archive named the archive...

Number of Entries: Maximum # of supported entries

Number Used: The number of files stored in the .t64 image. There is one instance where I found it to be wrong, and it is in older .t64 files. (see .t64 notes)

Version: Version of the .t64 tapeimage.

Bytes: Number of bytes in the file.

Type: This will be ---, PRG, or IMG. --- stands for an empty space in the directory.

1.17 .t64 Notes...

.t64 Notes

One has the header C64 tape image file, and is an older tape image file. These older images may have some problems with them, one of which is that the filesize in some of them can be wrong, and the other is that some of them may claim that there aren't any entries, but still contain one. Another problem is that they use a filetype #68 for PRG files. 1 is PRG, and 2+ are IMAge files. This program will STILL check for files even if the archive says there aren't any in it.

1.18 .p00 Support

UNd64 v 38.18 (12.1.98)

© Copyright 1994-1998 Jess Sosnoski

WHAT ARE .p00 PROGRAM IMAGES?

.p00 files are a format used on the PeeCee C-64 emulator C-64s, they are basically a C-64 program with a header on front with an identifier and the C-64 filename. I guess this format was created to make storing C-64 files on a PeeCee a bit easier, since the C-64 has 16 character-long filenames.

.p00 Output

UNd64 38.18 (12.1.98)
© Copyright 1994-97 Jess Sosnoski

Size Filename

33247 logo

Not Much to Explain here...

It was brought to my attention that some .p00 files may have a corrupt file starting address (=0), and there is a program on aminet to correct this...it's called fixp00.

I have added an option to und64 to catch these instances of .p00 files and correct them.

If you attempt to process a file with a start address of 0, the following will appear:

```
*Error: File address =0
```

If you were extracting the file, you would then be presented with the following options:

(F)ix, (I)gnore, (A)bort:

* Fix would set the file's address to \$0801 (#2048), which is the normal address for basic programs to load into on a c-64. Attempting to directly load a program which has a starting address of 0 on a c-64 would very likely crash the machine in a fatal way. (i.e. load"joe",8,1)

It is possible that the file's address actually might not be 0 or \$0801, but another address...and I will be adding an option in a future version of und64 to specify an address if you wish to do so.

* Ignore would simply write the file out..as-is.

* Abort would act as if you pressed ctrl-c to stop the program.

1.19 .lnx Support

UNd64 v 38.18 (12.1.98)
© Copyright 1994-1998 Jess Sosnoski

WHAT ARE .LNX ARCHIVES?

LynX archives are created by a program written by Will Corley for the C-64. They are basically the merged contents of a diskfull of files.

This program supports all .lnx archives, all varieties.

Lynx 9 and less notes:

Now lynx versions less than 10 are supported, however, some older lynx files may have an improper file length on the last file, UNd64 will attempt to correct this.

.lnx OUTPUT

UNd64 38.8 (27.5.96)
© Copyright 1994-97 Jess Sosnoski

Archived With: lynx xv by will corley
Number of Entries: 2

Size	Type	Filename
27317	SEQ	lynx xv.docs
15744	PRG	library 9.0

Archived With:
Will give you the version of Lynx that the archive was created with,

Type:
Same as those in .d64's, however, it will attempt to extract REL files. I don't know what will happen in this event, so if someone runs into one, let me know what happens.

1.20 .lbr Support

UNd64 v 38.18 (12.1.98)
© Copyright 1994-1998 Jess Sosnoski

WHAT ARE .LBR ARCHIVES?

LiBRary archives are created with a program on the '64 called...Library! They are similar to Lynx files. This program supports the uncompressed version of these archives.

Output is similar to .lnx files

1.21 .sfx support

UNd64 v 38.18 (12.1.98)
© Copyright 1994-1998 Jess Sosnoski

WHAT ARE .SFX ARCHIVES?

SFX archives seem to exist for several types of machines, including the '64! They are Self-eXtracting archives. You COULD run them on your c-64 or emulator to extract the files and wait for them to decompress, OR...you can decompress them nice and FAST on your Amiga!

This program will convert them to a convenient lzh/lha-compatible archive. (Then, you can use lha to do what you wish with them)

You can dig up a few of these files at ftp.funet.fi.

Listing and testing serve the same function, and will show the output file's name & size.

1.22 Progress Messages

Progress Messages

***Break

You pressed CTRL-C and stopped execution.

>>Extracting/Testing Errors:

* Not a recognized archive!!

- a.) The filelength wasn't EXACTLY 174848 or 175531 bytes, which all legal 35-track .d64's are.
- b.) The file's header didn't contain any of the legal archive identifier strings.

* You must decompress this file with UNZIP first!

* C-64 zipcode files are not the same as PC .ZIP files!

Oops! And you thought that they had .ZIP for the C-64 (*SLAP*)

Look

HERE

* Circular Link Found, File Truncated

UNd64 found a circular link in a .d64 archive and cut the output file off. A circular link occurs when a pointer to the next track & sector of a file points to a track & sector earlier in a file, which can cause an infinite loop (not good).

* Circular Link Found in Directory

Same as above, except it's in the directory structure. UNd64 will stop going thru the directory if it finds one of these nasties.

* File size error (n bytes)

In a .d64 file this is either caused by circular links, improper file size in the directory block, or a corrupt file (possible disk error). In a .t64 file this is caused by an improper file length stored in the directory header. I.E. if the filelength + file offset > .t64 filesize.

The number in parenthesis is the number of bytes the file actually was.

* Illegal Track or Sector

Relax, there's nothing wrong with your Amiga's drive, this just means that a link to a track and sector in a file on the .d64 diskimage is corrupt (probably xferred from the original disk). And most likely (or absolutely likely) a 15x1 drive would complain the same way if you tried to access the file in question from the original disk.

* Illegal Track or Sector in Directory

Same as above, except that it was in the directory structure instead of a file structure.

* File Exists! (R)ename (O)verwrite (S)kip (A)bort

If a file already exists with the same name, this message appears. At this point you would select the first letter of the appropriate option and press return. Rename will ask you for a new name to give the file, Overwrite will overwrite the file, Skip will skip extracting the file, and Abort will stop the program A-la CTRL-C.

* Not a 35-track 1541/1570 .X64 image!

This happens if you try to process an unsupported .x64 diskimage.

* Files pointer points past EOF

Means somewhere along the line, the archive got cut short!

*** Cant Read Input File!

*** Cant Write Output File!

*** Couldnt Allocate Memory!

*** Error Reading Input File!

*** Error Writing Output File!

*** Corrupt Archive!

These 6 errors are all fatal, and the program will exit immediately.

They're pretty self-explanatory when they happen.

(There's almost always a guarantee that the Amiga will throw a requester of some kind at you before one of the above I/O errors)

1.23 Compatibility

COMPATIBILITY

This program was developed on:

The Former Amiga 600 from hell

Tested on:

A600 2Chip 1.3/1.3

A600 2Chip 4Fast 2.05/2.1

A600 2Chip 4Fast 3.1/3.1

A600 2Chip 8Fast 3.1/3.1 (020/882/25)

A600 2Chip 16Fast 3.1/3.1 (030/882/33)

A1000 512K Chip 1.1,1.2,1.3

A3000 2Chip 8Fast 2 16BitFast 2.04/2.1, 3.1/3.1

(25 mhz '030 with LOTS of SCSI devices on it)

...And it worked on all of them, but bug reports are still welcome.

1.24 History:

HISTORY

- 37.1 First Release. Written and compiled in PCQ Pascal 1.2d (the PD release), assembled on A68k 2.71, disassembled with IRA 1.02, re-assembled and optimized with PhxAss 3.30, linked throughout with PhxLnk 2.03, and hunkmerged with Imploder 4.0.
Basically, it does what is says it does.
It has a styleguide-compliant version string.
(or seems to be)
Size: 9340 (19.4.94)
- 37.2 Added circular-link checking and fixed a ctrl-c bug when attempting to break while extracting (37.1 didn't stop). Unfortunately c-l checking slows down extraction a wee bit. Fixed REL file bug, it no longer lists or attempts to extract these nasty little files. Improved illegal filename character checking. Rewrote some parts of this doc.
Size: 10080 (21.4.94)
- 37.3 Speeded up circular-link check to take constant time $O(1)$, instead of summatory time, improved output. Added duplicate filename checking. Improved illegal filename character checking. Rewrote some more of the doc.
Size: 10684 (27.4.94)
- 37.4 Now primarily assembled with SNMA 1.39 instead of A68K, and re-assembled & optimized with Phxass 3.71, producing shorter faster code. Added ability to process deleted and splat files, and put some output filesize checking in.
Size: 9944 (4.5.94)
- 37.5 Assembled with SNMA 1.95, and skipped Phxass because it's optimization was causing GURU's-and I didn't want to mess with the program to get it to work at 3am. Added file testing, support for locked, etc.. files, removed a few bugs, rewrote some parts of code. Improved output.
Size: 11888 (10.6.94)
- 37.6 Assembled and optimized with SNMA 1.97. Rewrote reading/writing routines for a MAJOR speed increase. Rewrote parts of doc.
Size: 12168 (3.1.95)
- 37.7 Added archive filesize checking in for almost perfect .d64 archive recognition.
Size: 13000 (28.1.95)
- 38.0 TOTALLY rewrote reading/writing routines, MUCH faster. Rewrote and cleaned up even more code, which fixed quite a few nasty bugs.
-

Added AmigaGuide documentation, with a text editor!
Fixed bug when file was improper length, now reports correct length. Fixed filename bug, now filenames aren't padded with spaces. Output redirection works again. Fixed benchmarks in the doc, more accurate results made by sending program's output to >NIL:.
Improved output. Now also lists (but ONLY lists) REL files. Now locked and splat files are marked as such. (i.e. all file attributes list just like an old 1541)
Beefed up error checking for all kinds of stuff.
Added .t64 and .p00 support, both types are auto-detected. Primarily assembled & optimized with SNMA 1.97, linked with PhxLnk 4.03, disassembled with IRA 1.02, reassembled and optimized even more with PhxAss 4.10.
Now compiled with the 3.1 PCQ Includes.
Size: 16028 (14.3.95)

- 38.1 Fixed a bug in the command parsing which required one to select another option in order to use the no filename conversion option.
Assembled with snma 1.99, reassembled with PhxAss 4.18, and linked with PhxLnk 4.17.
Size: 15960 (5.7.95)
- 38.2 Rewrote fileexists function, parts of the filesize function, file extraction code, directory track parsing code, and track & sector link parsing code.
File sector parsing now reports illegal tracks/sectors. Dir track parsing code now checks for illegal track/sectors as well as circular links.
Added output path. (something useful in a dearcng utility)
Added more types of error messages.
Assembled with snma 2.03, reassembled with PhxAss 4.23.
Size: 19692 (24.10.95)
- 38.3 Fixed a bug in the filesize function which was not freeing up a fileinfoblock in certain situations. Also fixed a possible bug in the fileexists function (if it wasn't a bug, it won't matter anyway due to the fix)
Cleaned up some ancient code and vars, and did a wee bit of optimization in some routines. (smaller code)
Size: 19600 (28.10.95)
- 38.4 Removed a bit of redundant code and cleaned up the filesize function (again! :<), seems that in the unlikely situation if it were able to lock the file, allocate a fileinfoblock, and then fail to examine it, it wouldn't free up the fileinfoblock...FIXED (finally).
Optimized bits'n'pieces of this and that, etc...
Assembled with snma 2.04 and phxass 4.25 now.
(smaller code than 38.2 or 38.3 !)
Size: 19376 (26.11.95)
- 38.5 Added dates to the history.
Now assembled with phxass 4.26 and linked with phxlnk 4.20.
Fixed (I hope!) mungwall hits and crashes when a long
-

pathname is used or long filename is used.
Now accepts up to 255 characters for filenames.
Size: 19372 (13.1.96)

38.6 Fixed the date of the last version in the docs.
(damn, I hate when I forget what year it is)
Added X64 .x64 diskimage support for 1541/1570 diskimages.
Now extracts .t64 IMAge files.
(if there's any problems, contact me)
Size: 20164 (17.2.96)

38.7 Modularized the code into separate drivers, optimizing
and cleaning up their code.
Reads in the archive all at once, then processes it.
Fixed 2 bugs in the .t64 driver: if the file's offset
was beyond the end of the archive the old driver wouldn't
catch it probably resulting in...your guess is as good as
mine!, and if a file couldn't be successfully written, only
the input file was closed. (fixed!)
Fixed a bug in the fileexists function, it ignored the
path, and only checked the current directory. (fixed)
Assembled with PhxAss 4.30, and linked with PhxLnk 4.23.
Size: 19940 (7.5.96)

38.8 FINALLY added .lnx support.
Did a couple teeny optimizations here and there too.
Assembled with phxass 4.32.
(okay...so I didn't change the date in a couple days :))
Size: 22636 (27.5.96)

38.9 About 300 bytes of optimization from removing some redundant
code.
Fixed a NASTY bug (DAMN, I swear it worked before I uploaded
it!) in the .d64/x64 routine that crashed when attempting
to list/test a diskimage.
Size: 22344 (20.6.96)

38.10 Minor update, added support for .d64 files with appended
error information (which is ignored). And a couple *small*
optimizations.
Size: 22236 (17.9.96)

38.11 Changed .d64 reading, listing takes 1 sec. longer, but
more reliable now (reads in whole image), added recognition
for another .t64 ID. Made reading of archives other than
.d64's more reliable by checking ID's first rather than
filesize. Took version restriction out of .t64, and Lynx
modules. Lynx module now supports punter-lynx archives.
Added .lbr Library support.
Added checking for corrupt .t64/.p00 archives.
Did a couple teeny optimizations here & there.
Overworked this guide a bit, fixing lots of
obsolete information.
Fixed a bug in the fileexists function when renaming a file.
Size: 23800 (10.10.96)

38.12 Added support for .sfx files, converts them to a lzh/lha

compatible archive.
Now using phxass 4.33 for assembling.
Size: 24516 (19.10.96)

38.13 Reordered and overworked this .guide.
Rewrote a good bit of the main program and .d64 module,
getting rid of that damn 180k+ BSS hunk...now it no
longer allocates 180k every time you load the program.
It only allocates what it needs when it needs it! :)
Also makes loading .d64's a couple milliseconds faster
because of the way memory is used now.
Now using PhxAss 4.34 and PhxLnk 4.30 in the make process.
Fixed a bug when the pointer to the next track & sector
pointed to track 0 sector 0, causing the program to report
a write error (fixed).
Removed checking for circular-links in the directory
structure for now. (sorry :()
Fixed an annoying bug in the .p00 module when testing
an archive...it sometimes reported that a file existed
even if not extracting (fixed).
Minor Code Cleanup.
Size: 24601 (10.2.97)

38.14 Reordered and cleaned up lots of code.
Small Optimizations.
Fixed bug in CBMASCII->ASCII conversion which did
not converting all characters to ascii.
Added second-pass of space-padding removal if conversion
of illegal characters generated spaces at end of filenames.
Added RLE-Compressed !!zipcoded split diskimage support!
Added fixp00 feature to the .p00 processor.
Added output of comments in .X64 archives.
Added the use of Thor software's HUNK program for further
processing of the und64 executable before distribution.
(smaller executable)
PhxAss 4.35 used during assembly.
Size: 25896

38.15 Recompiled under the latest uploaded version of PCQ
pascal on Aminet.
Now assembled with PhxAss 4.37beta.
Now uses some 2.x functions instead of home-made ones.
Replaced StrtoInt function with one from dos.library
Removed buggy custom stringcopy function, fixing
major crashes it was causing on some machines.
Added some optimizations in zipcode handler.
Changed the way the program processes paths.
Size: 25984 (3.7.97)

38.16 Now Linked with PhxLnk 4.31alpha.
Links with amiga.lib before linking with pcq.lib resulting
in a smaller exe size.
Executable now has os 2.x short-rel hunks.
Size: 24936 (9.9.97)

38.17 Linked with PhxLnk 4.31, Assembled with
PhxAss 4.37b (24.10.97).

Compiled with PCQ Pascal 2.0 recently uploaded to Aminet.
CHANGED a good bit of internal stuff to take advantage of
all the new stuff in the pcq 2.0 includes, this also
involved removing old routines from my code that did
the same things.
Only released on my webpage in beta form, had some
problems with the new PCQ 2.0 stuff.
Size: 24860 (30.10.97)

38.18 Went back to custom routines because the bonus ones from
PCQ 2.0 weren't 100% compatible with the way I used mine.
Assembled with PhxAss 4.38alpha.
Also added a file to append to your s:filetypes file
for the whatis.library for various cbm archives and
executables. (All's we need now are icons for 'em!)
Added recognition for PeeCee .ZIP files so that und64
now warns the user that c-64 zipcode files aren't the
same as PeeCee .ZIP files.
Size: 26376 (12.1.98)

1.25 TO DO...

TODO

-Rewrite this guide
-REWORK the entire source to take FULL advantage of the PCQ 2.0 stuff.
-Rest of Todo list is now internal, all sorts of interesting top secret
stuff. ;>

1.26 Author/Updates...

The Author's Addresses...

Jess Sosnoski
651 Hillside Drive
Mount Carmel, PA 17851-2463
USA

Email:
starblaz@postoffice.ptd.net

IRC Nick: starblazr
usually on:
Dalnet: Usually in #nin , #miggy1, #amiga or other amiga channels.

Galaxynet: Could be in any amiga channel

Pulsarnet: #elitecafe

Updates can be found on my webpage, and are usually put there before
they appear in Aminet's RECENT file.

WWW:

<http://home.ptd.net/~starblaz/>

If you feel the irresistible need to send me money, gifts, cigarettes or an Amiga 4060T, I will give you a very gracious thank you.

```
  .  
 /|  
/_|  
_/ _|MIGA...Mooooo!
```

1.27 And...The Quote of the Day:

Quote of the Day:

```
"xX dIGITAL CORRUPTION xXx Legal Coding Division Xx"  
  -The eternally cool bots of #elitecafe on Galaxynet
```

1.28 The Amiga 600 Formerly From Hell

The Former Amiga 600 From Hell:

Kickstart/Workbench 3.1

(Kickstart 40.63 (A500/2000/2500 version), and works PERFECTLY!)

Apollo A630 68030/68882 33Mhz Accelerator

16 MB 32-Bit Fast ram

2 MB Chip

120MB Conner 2 1/2" internal IDE HD

DataFlyer SCSI+ 1200 Scsi interface

DataFlyer XDS External IDE interface With a JTS Palladium 1 GB IDE HD

Sony CDU-590 Double Speed SCSI CD-ROM

Hayes Accura 288 V.FC + FAX

Hewlett Packard DeskJet 540c

Amiga 2002 Monitor (A wee bit nicer than the old 1080, but not much)

Amiga 1010 low-density external floppy.

1 Heavy rubber band to hold the case together

It WAS the '600 from hell until I found a guy that had a A600 TOWER!

So I guess now my A600's just hangin' out by the river Styx!

oh, yeah, and my case will never close quite the same way again!

1.29 Thanks to...

Thanks to:

Kay Drangmeister

For bugreports, and sourcecode for readargs, and testing it on an '060.

Jouko Valta
Author of X64, for sending me some .x64 files to test with UNd64.

Michael P. Steer
For sending me some sourcecode for zipcoded files.

Johan "S.Duvan" Alfredsson and Henrik "Puh" Nilsson
of Oepir Risti Translations (d95duvan@dttek.chalmers.se) (alvar@one.se)
for the Swedish translation of this guide!

Mitch B. Parker
For his inspration which is responsible for me creating UNd64 since
the first version.

Dave Pedergnana
Without whom, I might never have become interested and involved with
computers....infinite thanks.

Amiga International and Gateway 2000
For re-re-resurrecting the Amiga!

Everyone else who sent me ideas, and/or bugreports.

Everyone who registered! :) (So far...28)

And...Everyone who spread the program.

If you have ever sent me mail, for bugreports/ideas,
and your name is not mentioned here, and you would like it mentioned here,
email

me

.
