

PufferFish

COLLABORATORS

	<i>TITLE :</i> PufferFish		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		September 19, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	PufferFish	1
1.1	main	1
1.2	What is PufferFish?	1
1.3	How to get help	2
1.4	How to enter floppy numbers	2
1.5	How to select floppy drives	3
1.6	How to start generating floppies	3
1.7	ToolTypes and command line options	3
1.8	Version information	4
1.9	Credits	5
1.10	Possible bugs and enhancements	5

Chapter 1

PufferFish

1.1 main

PufferFish V1.0

What is PufferFish?

How to get help

How to enter floppy numbers

How to select floppy drives

How to start generating floppies

ToolTypes and command line options

Version information

Credits

Possible bugs and enhancements

1.2 What is PufferFish?

PufferFish is a program to uncompress the archived Fish disks ↔
stored

on the FreshFish, FrozenFish and GoldFish CDs. It came about as a result of a Usenet posting by Fred, who was looking for someone to write just such a program.

To deal with messages between the interested Amigans, a mailing list known as GenFloppy was created. (In fact, that was the original name of the program. The name PufferFish came from my mind after a long night of programming. 8^) The program you have now is a result of feedback from that list. The most notable features requested were:

- * Runs on 2.x and above (only)
- * Selection of a specific floppy or range of floppies
- * Progress report (seeing the output of each program)
- * AmigaGuide help (context-sensitive under 3.x)
- * Source provided and freely redistributable
- * Formats disks, finds, unpacks and verifies archives
- * Several different floppy number input methods
- * Uses only selected floppy drives
- * Allows one (configurable) additional device, such as RAD:
- * Configurable format, unarchive and verify archive commands
- * Configurable first/last disks
- * Configurable contents path

The only notable exception is that Release 2.04 (V37) or higher is required. (Sorry, Ethan!)

In the original spirit of the library (and in case I get hit by a meteor from the Euphrates sector), I have included the source code. It's by no means perfect (although all known bugs have been squashed), but it does provide a very elementary example of using BOOPSI gadgets and tasks. (I'd like it to be a more thorough example, but I don't know how to use a lot of BOOPSI myself! See
enhancements
.)

Please send me any changes that you might make (especially BOOPSI-related ones). Also, while the program and source are freely-distributable, they're still © 1994 Peter Janes.

Any comments or suggestions are welcome.

Peter
pejanes@descartes.uwaterloo.ca (preferred; valid until at least April 1995)
pejanes@io.org

1.3 How to get help

Press the Help key. That's what it's there for, after all!

8^)

Under 3.x and above, help is context-sensitive; users of previous OS releases will be presented with the main help node.

1.4 How to enter floppy numbers

Floppy numbers may be entered using four different methods.

- 1) Type the numbers into the integer gadgets. Pressing 'Tab' will activate and switch between the integer gadgets.
 - 2) Slide and/or click the slider until the proper number is shown.
-

- 3) Click the arrows and digits until the proper number is shown.
- 4) Use the keyboard equivalents for the slider. The equivalents for the top slider are '-' and '='; for the bottom slider, '_' and '+' (Shift + '-' and '=').

Note that the ending disk number must be greater than or equal to the starting disk number. This is enforced by the program. It's also an easy way to set ranges, since setting the starting number also increments the ending number if necessary.

1.5 How to select floppy drives

There are two ways to select floppy drives.

- 1) Type the appropriate digit on the keyboard or keypad to toggle a drive. (If an additional device was configured, use '4' to select it.)
- 2) Click the checkbox gadget corresponding to the appropriate drive.

1.6 How to start generating floppies

Click the "Go" button or type 'G'. Feed the floppies into the appropriate drives and press Return (in the appropriate window) to start the format.

1.7 ToolTypes and command line options

FIRSTDISK

Sets lowest disk number.

LASTDISK

Sets highest disk number.

CONTENTSPATH

Sets path to contents file.

FORMATPATH

Sets command to use for formatting disks. Defaults to "Format NOICONS".

LHAPATH

Sets command to use for expanding disks. Defaults to "LhA -a x".

BRIKPATH

Sets command to use for verifying disks. Defaults to "Brik -Cb".

VERBOSE

Enables echoing of commands. Useful for debugging non-default commands.

DEVICE

Sets the single user-configurable device. Useful for disk librarians--you can quickly generate a single disk image on RAD: or another fast device and then DiskCopy'ing it to floppies.

IMPORTANT: the ':' must be included at the end of the device name. Also, the program attempts to format each device it uses, so don't try generating floppies to GIG:, your 1.5 gigabyte hard drive.

1.8 Version information

This is PufferFish V1.0, the first public release.

Beta release history

Changes from V0.3:

- * New icon, courtesy of Walter Doerwald
- * Disables window input when error requesters displayed
- * Now uses a slightly modified version of Doug Keller's ButtonClass (from ButClas3.lha). Search for // PEJ in ButtonClass/ButtonClass.c. This gives that nice-looking disabled-gadget imagery that the generic BOOPSI class won't. (My change was, basically, to move the outline-drawing code together so that selected checkmark gadgets wouldn't overwrite the shine part of the outline.) Also uses ButtonClass for "Go" gadget.
- * '4' shortcut for extra device now works.
- * Additional testing with multiple devices. It works! How about that... there appear to be no bugs in the code the first time around....
- * Unless brik changes sometime soon (it's lasted this long, so it probably won't), a 20000-byte stack works fine.

Changes from V0.2:

- * Added user-configurable device through DEVICE ToolType and CLI parameter.
- * Added wb2cli.o. This will solve the problems with paths for commands.
- * Removed (commented) CloseLibrary(DiskBase) from source--don't do that!
- * Moved input handling to a separate routine
- * Requester displayed if use requested help but amigaguide.library not available

Changes from V0.1:

- * Improved handling of numeric gadgets
 - * Configurable first/last library disks
 - Defaults to 1/1000
 - Can handle up to 9999 disks
 - * Configurable contents file path
 - Defaults to Contents, in current directory
 - * Configurable commands/command paths
 - Defaults to "Format NOICONS", "LhA -a x" and "Brik -Cb"
 - * Runs if amigaguide.library is not present
 - * Runs from CLI when no arguments specified
-

- * Added VERBOSE ToolType/argument
- * Output windows are offset
- * Contents file parsing now handles quoted disk names
- * Default contents name changed to PufferFish.db
- * Main window disabled and busy pointer set while generating floppies
- * Stack size for 'brik' upped to 20000
- * Integer gadgets have boxes
- * Neat new name! 8^)

1.9 Credits

Thanks to:

- * Walter Doerwald for the neat PufferFish icon
- * Sebastiano Vigna for his code in files.c
- * Ethan Dicks for his comments (still interested in a 1.3 version after seeing the code? 8^)
- * Mathias Axelson for his comments
- * Mike Stark for his BOOPSI class "BoxedStringClass"
- * Doug Keller for his BOOPSI class "ButtonGadgetClass"
- * Michael Sinz for WB2CLI.o, and Carolyn Scheppner and CATS for making it freely-distributable
- * all the other unnamed beta-testers on the PufferFish mailing list
- and, of course,
- * Fred Fish and all the contributors to the Amiga Freely-Distributable Software Library

1.10 Possible bugs and enhancements

If anyone has any suggestions on how to fix/enhance the following, or if they work/don't work, please let me know!

- * Automatic regeneration of failed disks when using ranges
Right now, just informs the user of the failure and continues to the next disk.

- * Non-flickering checkboxes
This happens because of my workaround for GA_ToggleSelect. The checkboxes are specifically set as selected or not selected after an IDCMP_GADGETUP message because the GA_ToggleSelect option doesn't seem to work for BOOPSI "buttonclass" objects. As a result, Intuition draws the unselected image before PufferFish draws the selected one.

This happens even with ButtonClass. A fix would be appreciated.

* Scalable checkboxes

Are scaled images even available from BOOPSI? intuition/imageclass.h seems to suggest so, but I've had no luck.

* Easier layout of gadgets and frames

This could probably be made much simpler using a "groupgclass" object, but I have no idea how to set one up. Anyone?
