

PDHFIC

Chris Young

COLLABORATORS

	<i>TITLE :</i> PDHFIC	
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>
WRITTEN BY	Chris Young	August 26, 2022
<i>SIGNATURE</i>		

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	PDHFIC	1
1.1	Welcome to PDHFIC...	1
1.2	DISCLAIMER	1
1.3	Technical Stuff - not for the faint-hearted	2
1.4	Introduction to PDHFIC	2
1.5	Distribution	2
1.6	Requirements	3
1.7	Installation	3
1.8	Usage Instructions	3
1.9	Quick Start	3
1.10	The Graphical User Interface (GUI)	4
1.11	menu-project	4
1.12	menu-picture	4
1.13	menu-settings	5
1.14	menu-snapwin	5
1.15	menu-autoname	5
1.16	submenu-mapcols	5
1.17	menu-mapcols-auto	6
1.18	menu-mapcols-ink	6
1.19	menu-mapcols-paper	6
1.20	menu-tzxcustomborder	6
1.21	menu-intellifilter	6
1.22	menu-progresswin	6
1.23	submenu-tzxblocks	6
1.24	menu-warnover	7
1.25	menu-writeheader	7
1.26	tzx-archiveinfo	7
1.27	tzx-standard	7
1.28	tzx-custom	7
1.29	menu-help	7

1.30	intui-zoom	7
1.31	intui-back	7
1.32	gui-inputselect	8
1.33	gui-outputselect	8
1.34	gui-romremap	8
1.35	gui-zxdtpal	8
1.36	gui-scalepic	8
1.37	gui-smooth	8
1.38	gui-saveas	9
1.39	gui-convert	9
1.40	menu-about	9
1.41	menu-sep	9
1.42	menu-quit	9
1.43	menu-info	9
1.44	menu-output	9
1.45	menu-advopts	9
1.46	menu-boredom	10
1.47	menu-clipboard	10
1.48	menu-display	10
1.49	menu-saveconfig	10
1.50	menu-screen	10
1.51	Command Line Arguments	10
1.52	Tooltypes	11
1.53	The Input File Formats	13
1.54	The Output File Formats	13
1.55	Future Plans	14
1.56	Thanks To...	14
1.57	Contact Information	15
1.58	Colour Mapping Algoritm	15
1.59	Who in their right mind is actually going to use this feature?	16
1.60	Example SCRs	16
1.61	Program History	17
1.62	Known Bugs (most important first)	18
1.63	The OS3 ROMREMAPping	18
1.64	The Competition Results	19
1.65	ARexx Port	19
1.66	arexx-convert	20
1.67	arexx-selectsource	21
1.68	arexx-selectdest	21

1.69 arexx-quickconvert	21
1.70 arexx-scale	22
1.71 arexx-noscale	22
1.72 arexx-romremap	22
1.73 arexx-stdremap	22
1.74 arexx-altpalette	22
1.75 arexx-zxdtpalette	22
1.76 arexx-smooth	22
1.77 arexx-nosmooth	23
1.78 arexx-brightsense	23
1.79 arexx-coloursense	23
1.80 arexx-whitesense	23
1.81 arexx-redyellow	23
1.82 arexx-bluecyan	23
1.83 arexx-greenmagenta	23
1.84 arexx-flashred	23
1.85 arexx-flashgreen	24
1.86 arexx-flashblue	24
1.87 arexx-version	24
1.88 arexx-saveformat	24
1.89 arexx-getsavformat	24
1.90 arexx-getsources	24
1.91 arexx-getdest	24
1.92 arexx-getbright	24
1.93 arexx-getcolour	25
1.94 arexx-getwhite	25
1.95 arexx-getredyel	25
1.96 arexx-getblucyn	25
1.97 arexx-getgrnmag	25
1.98 arexx-getflashred	25
1.99 arexx-getflashgrn	25
1.100 arexx-getflashblu	25
1.101 arexx-getscales	26
1.102 arexx-getsmooth	26
1.103 arexx-getaltpal	26
1.104 arexx-getromremap	26
1.105 The ARexx Scripts	26
1.106 Frequently Asked Questions (which I largely just made up)	26

Chapter 1

PDHFIC

1.1 Welcome to PDHFIC...

PDHFIC 2.5 by Chris Young and Joe Mackay

© 1998 Unsatisfactory Software

DISCLAIMER

[Introduction](#) [Distribution](#) [Requirements](#) [Installation](#)

[Instructions](#) -> Including quick start!

[Technical Details](#) File formats, colour mapping, FLASH mode etc...

[ARexx Port Example](#) [ARexx Scripts](#) [Example SCRs](#)

Past: [Program History](#)

Present: [Known Bugs](#)

Future: [Future Plans](#)

[Thanks to...](#) [Contact Information](#) [Competition Result](#)

[Frequently Asked Questions](#)

1.2 DISCLAIMER

PDHFIC is provided "AS IS", without warranty of any kind either expressed or implied. It has proven to be stable in everyday use, but even so you still use this program at YOUR OWN RISK. I will not accept responsibility for loss of data, damage or anything else caused either directly or indirectly by PDHFIC.

Modified copies of PDHFIC must not be distributed. I have purposely not included the source code in this archive for that reason*. Please don't ask for it unless you have a very good reason (eg. if you think I've discontinued development, and want to take the project over or something). I have no problems if somebody wants to do a port of this, but be warned that;

a) it isn't written in C

b) it makes heavy use of Amiga ROM functions

In fact, you would probably be better off trying to port ppmtoscr. Although that isn't written in C, either. I do have an old "half converted to C" version of it, but it doesn't create valid SCRs. Er.. where was I?

PDHFIC is not in any way based on the source code for Blood's BMP2SPEC program.

* The fact that the source is in a complete mess, with "comments" (ie. old bits of code) floating around and that even I have trouble understanding what's going on has got absolutely nothing to do with it. Probably.

Bad programmers, like good chefs, never reveal their sources (sauces).

Sorry, I meant *good* programmers. Tch, eh?

1.3 Technical Stuff - not for the faint-hearted

[Input File Formats](#) and [Output File Formats](#)

[Standard Colour Mapping](#) and [ROMREMAP Mapping](#)

[FLASH Mode](#)

1.4 Introduction to PDHFIC

OK. You're probably all wondering what PDHFIC actually does. In fact, half of you are probably wondering what PDHFIC stands for. In which case I'm going to disappoint you and say nothing.

PDHFIC converts Datatypes images to the Spectrum formats: SCR, TAP, TZX, ZX82 and .header/.bytes. It can scale or truncate images so they are the correct size.

But why did I create this? What about PPMtoSCR? Didn't I port BMP2SPEC over to the Amiga?

Basically, I had been wanting to create Spectrum format images for quite some time, when I heard about BMP2SPEC. Feeling interested (and accidentally stumbling across it while browsing through the World of Spectrum archive), I discovered that the source code was included. Not one to refuse a challenge* I attempted a re-compilation of it. Almost suffering from a major heart failure when it compiled first time with no modifications, and tearing my hair out wondering why it worked perfectly well one minute and crashed the next, I got through at least two different compilers and a hot (or, at least, it was when I started) mug of strong coffee before working out what the problem was.

Needless to say, I decided that BMP2SPEC was a bit limited (not least because it required BMP files of exactly the right size for input) and lacking in features. So I wrote a little NetPBM extension called PPMtoSCR.

Expanded over several years (well, months), the program had built-in routines to call other NetPBM programs and a "sort of" GUI. It really was getting a bit far away from its filename.

So, when Joe Mackay (Hi Joe!) offered to send me the Datatypes code he was going to write for a DT2SCR type program, I saw the opportunity to change the program beyond recognition. And split it up into two parts in a confusing way, which I would never be able to explain in the docs. In fact, I appear to have confused at least a couple of people already.

So, I added a GUI. Realising that real men use GadTools** (MUI is for wimps or, more accurately, WIMPs), that was exactly what I used.

PPMtoSCR still exists, but in a cut-down form - PDHFIC is more of a successor to PPMtoSCR than the new ppmtoscr 3.0 is. In effect, PDHFIC is the image convertor, and ppmtoscr is a small CLI-only program which will interface with NetPBM quite nicely (well, that was the whole point).

Features of PDHFIC;

Too many to list...

* The challenge being whether I, with (at that time) hardly any knowledge of C whatsoever, could convert a program written in C. Ahem.

** Actually, real men probably don't bother with programming and decide to take up fighting or something instead.

1.5 Distribution

You can distribute this as much as you like, but please note the following permissions and restrictions;

You must distribute this documentation with PDHFIC. Everything else included in the archive does not need to be kept with the main program.

Aminet, NVG and World of Spectrum are perfectly entitled to distribute PDHFIC through FTP. Everyone else is allowed to do that as well, providing the file is accessible by **everybody** at **no cost** (ie. Anonymous FTP).

You can put it on a disk and distribute it like that if you want to, but PDHFIC is so small that it should really be included with a few other programs. Again: No charge, except for the price of the disk!

Distribution on CD-ROM is perfectly OK. If you are not mentioned below and are thinking of distributing PDHFIC on a CD made for profit, then you must ask me first. I would appreciate a copy of the CD in question, as if you are making money out of my program, then I believe I should also get something out of it. If you don't agree with this, then don't put it in a compilation.

The following people have my permission to include PDHFIC on their CDs. They are not required to notify me or give me anything, although gifts won't be refused :-)

1. Aminet
2. Magazines (cover CDs/disks only - not special subscriber disks)
3. Gerard Sweeney

1.6 Requirements

PDHFIC uses several features of OS3.0, so you will need an Amiga running at least Workbench 3.0

Optionally required is;

als.library [ftp://sunsite.doc.ic.ac.uk/aminet/util/libs/als_lib.lha]

For [boredom relief](#)

SCR2GIF [<ftp://ftp.gns.getronics.nl/pub/os/sinclair/tools/amiga/scr2gif.lha>]

To [create GIFs](#)

Also recommended is the ZX Datatype, available on Aminet. Especially if you want to convert SCRs back to SCRs just to see how well PDHFIC works...

1.7 Installation

You've got "Installer", right? Well, simply double-click on Install.

1.8 Usage Instructions

I have divided this section of the guide up into several parts;

[Quick start \(GUI usage\)](#)

[Gadgets and Menus](#)

[Command line arguments](#)

[Workbench tooltypes](#)

1.9 Quick Start

To convert pic.iff to pic.scr, do the following;

1. Load PDHFIC - the GUI will appear.
2. Click on "Select" next to "Input", and choose pic.iff from the requester

If [Automatic Naming](#) is on, then the output file will automatically set to pic.scr in the same directory.

3. Select the desired output format from the "Save as" cycle gadget. For testing purposes (and if you have the ZX Datatype), I recommend using either SCR or ZX82.
 4. Click on "Convert!". The conversion will be performed. Try loading the resulting file into Multiview or a Spectrum emulator and admire the results.
-

1.10 The Graphical User Interface (GUI)

If you run PDHFIC from Workbench, then it will open a window looking absolutely nothing like the following; (don't you just hate it when people draw crap ascii representations of their program's GUI?)

```
Project _ Picture _ Settings _ Help _____ []
```

```
_____
[] _____ [] []
```

```
||
```

```
| Input: _____ Select |
```

```
| Output: _____ Select |
```

```
||
```

```
| Smooth Scale Pic |
```

```
||
```

```
| ROM Remap ZXDT Palette |
```

```
||
```

```
| Save as | SCR Convert! |
```

```
-----
```

(possibly followed by the **advanced options**)

1.11 menu-project

```
Project
```

```
About...
```

```
-----
```

```
Quit
```

1.12 menu-picture

```
Picture
```

```
Select Source...
```

```
From Clipboard
```

```
Display...
```

```
-----
```

```
Select Dest...
```

```
Display...
```

```
-----
```

```
Information...
```

1.13 menu-settings

Settings

Advanced Options

Use Screen

Warn Overwrite

Intelligent Filtering

Automatic Naming

Progress Window

Map Extra Colours To »

TZX Blocks »

TZX Custom Block Border »

Write .header

Snapshot window

Save Config

1.14 menu-snapwin

If this option is checked, the current window position will be saved along with the configuration (ie. check this, then select **save config** to snapshot the window, uncheck it then select **save config** to unsnapshot it.)

Default set by: WINX and WINY

1.15 menu-autoname

This toggles whether PDHFIC should automatically name the output file as follows;

If there is no output file selected, it will set it to the source directory and filename, but with an appropriate extension.

If there is already an output file selected, the directory will not be changed, but the filename will be set to the source filename and appropriate extension.

When changing the output filetype, the output filename's extension will be changed automatically (or added if one does not already exist)

Default set by tootype NOAUTONAMING

1.16 submenu-mapcols

Map Odd Colours To

Auto Ink

Paper

1.17 menu-mapcols-auto

Currently unavailable - use the smooth checkbox for now.

This option will toggle smooth mode, when I work out exactly how to implement it (I want to MX it with the ink/paper, but don't actually want to lose the current ink/paper setting. That's why I've disabled this for now...)

1.18 menu-mapcols-ink

This will tell PDHFIC to map all extra colours in each block to the INK colour.

Default set by tootype ODDCOLOURS=INK

1.19 menu-mapcols-paper

This will tell PDHFIC to map all extra colours in each block to the PAPER colour. This may cause "disappearing pixel syndrome"...

Default set by tootype ODDCOLOURS=PAPER

1.20 menu-tzxcustomborder

Allows you to set the colour of the border in TZX custom SCREEN\$ blocks

Default set by tootype BORDER

1.21 menu-intellifilter

This toggles whether to apply "intelligent" filtering to the input file requester. This will only show valid pictures, however it does slow down the reading of large directories. If the input file requester hangs your system, try disabling this option.

Default set by tootype NOFILTER

1.22 menu-progresswin

This toggles whether the progress window should open.

Default set by tootype NOPROGRESSBAR

1.23 submenu-tzxblocks

TZX Blocks

[Archive Info](#)

[Standard](#)

[Custom](#)

Defaults set by tootype TZXBLOCKS

1.24 menu-warnover

This will set whether PDHFIC should bring up a requester if you have selected an existing file to write over.

Default set by tooltype WARNOVERWRITE

1.25 menu-writeheader

This sets whether PDHFIC should create a corresponding .header file when saving in .bytes format.

Default set by tooltype NOHEADER

1.26 tzx-archiveinfo

This will set whether or not to write an "Archive Info" block to TZX files. The archive info will contain the author, copyright and annotation fields from the original file.

1.27 tzx-standard

This is a toggle switch which will set PDHFIC to writing "Standard" (ie. Spectrum normal speed) blocks.

1.28 tzx-custom

This is a toggle item which will set PDHFIC to writing a standard custom block of type 'Spectrum Screen'.

I know of no programs at all which can display these, so perhaps the author of ZX Datatype might like to implement it...?

1.29 menu-help

Help

[Selected Output Format...](#)

[Relieving Boredom...](#)

1.30 intui-zoom

This will shrink PDHFIC down (or up, rather) to its titlebar.

1.31 intui-back

This is ridiculous. You do actually have an Amiga, don't you?

1.32 gui-inputselect

Clicking on this will bring up a standard file requester allowing you to choose the source image. Please note that this requester will only display valid images - if it takes a long time to read directories and/or displays files which aren't pictures, then you may have a datatype (or two) which has trouble identifying pictures. This is not a bug in PDHFIC. Nor is it a bug in the datatypes, the problem is that some filetypes are very difficult to identify.

The main culprits are:

MacSND (takes a long time to identify files - it seems to have to read almost all the file)

MacPaint (often identifies files as pictures, when they aren't)

If the file selected is not a valid Datatypes image, a requester will appear to warn you.

Also note that PDHFIC's window is an AppWindow, so you can just drop icons onto it rather than going through all the file requesters.

1.33 gui-outputselect

This allows you to set the output filename. The file extension defaults to whatever you have selected as the output format.

WARNING: PDHFIC does not check to see if the output file already exists, and will overwrite it if it does.

Note that when saving .bytes files, a corresponding .header file will also be created, and overwrite any existing file of the same name.

1.34 gui-romremap

Use OS3's built-in colour remapping functions. See [OS3 Mapping](#)

Default set by tooltype ROMREMAP

1.35 gui-zxdtpal

In ROM Remap mode, this sets whether PDHFIC should use a palette identical to ZX Datatype's or a different one. See [OS3 Mapping](#)

Default set by tooltype ALTPALETTE

1.36 gui-scalepic

This sets whether to scale the picture or truncate it.

Default set by tooltype NOSCALE

1.37 gui-smooth

If switched on, odd pixels will be mapped to the nearest colour (INK or PAPER), when off, odd pixels will all be mapped to INK.

Default set by tooltype NOSMOOTH

1.38 gui-saveas

The cycle gadget sets the **output format**

Default set by tooltype SAVEFORMAT=SCR|TAPIZX82|BYTES|TZX

1.39 gui-convert

Actually carries out the conversion. Hurrah!

1.40 menu-about

Displays an about requester. As you would expect, really.

1.41 menu-sep

This does nothing, as it's just a separator. Twit.

1.42 menu-quit

Er...

1.43 menu-info

Shows information about the currently selected source image. It usually just shows the complete path and filename and the filetype, but after conversion will tell you whether the picture was scaled or truncated.

1.44 menu-output

Shows some information about the currently selected **output format**

1.45 menu-advopts

Shows/hides the Advanced Options panel, featuring sliders for the following colour mapping options.

The first three options are opposite to what you might expect, but sort of work like the controls on the telly. The last three work (sort of) a bit like the tint setting on really old TVs.

Please note: These do not affect the **ROM remapping** .

Bright: More BRIGHT <-----> Less BRIGHT blocks

Colour: More colour <-----> Less colour (greyscale)

White: More contrast <-----> Less contrast (less white)

Red/Yel: More red <-----> More yellow

Blu/Cyn: More blue <-----> More cyan

Grn/Mag: More green <-----> More magenta

Please see [tooltypes](#) and [colour mapping](#) for fuller details.

There is also a panel to select the FLASH colour. If any component is set to -1, then FLASH will be switched off. See [FLASH mode](#) for more information.

Defaults set by tooltypes: ADVANCED,FLASHRED,FLASHGRN,FLASHBLU, BRIGHTSENSE,COLOURSENSE,WHITESENSE,R

1.46 menu-boredom

Offers help on relieving boredom ;-)

If this is ghosted, then you should get the archive util/libs/als_lib.lha from Aminet.

1.47 menu-clipboard

This will cause PDHFIC to read an image from the clipboard, rather than from a file on disk. The clipboard unit number can be set with the CLIPUNIT [tooltype](#)

1.48 menu-display

Launches MultiView* to display the source or destination pictures, if they exist.

* I think it's pretty safe to assume all WB3 users have MultiView, isn't it?

1.49 menu-saveconfig

Saves the current configuration into PDHFIC's [tooltypes](#)

1.50 menu-screen

Sets whether or not MultiView should display pictures on a separate screen.

Default set by tooltype: SCREEN

1.51 Command Line Arguments

Shell Template:

DTFILE/A,SCRFILE,FORM=SAVEFORMAT/K,COL=COLOURSENSE/K/N,BRT=BRIGHTSENSE/K/N,WHT=WHITESENSE/K/N, BLUCYN=BLUECYAN/K/N,REDYEL=REDYELLOW/K/N,RED=FLASHRED/K/N,GRN=FLASHGREEN/K/N,BLU=FLASHBLU/NOSMOOTH/S,OS=ROMREMAP/S,ALTPAL=ALTROMPALETTE/S,NOHEADER/S,CLIPBOARD/S,NOCFG=NOCONFIGFILE/S

DTFILE:* This is your source image file

SCRFILE: This is the path and filename of the resulting file. If not set, it will use [auto-naming](#) to determine the output filename.

SAVEFORMAT: Specifies the output format. See [output formats](#)

COL=COLOURSENSE/K/N,BRT=BRIGHTSENSE/K/N,WHT=WHITESENSE/K/N,GRNMAG=GREENMAGENTA/K/N,BLUCYN

Colour mapping options. See [colour mapping](#) and [tooltypes](#)

RED=FLASHRED/K/N,GRN=FLASHGREEN/K/N,BLU=FLASHBLUE/K/N

Set when to use the FLASH attribute. See [FLASH mode](#)

NOSCALE: This will stop PDHFIC scaling your source image.

GREYSCALE: This stops PDHFIC from outputting colour images

NOBRIGHT: This prevents PDHFIC from using the BRIGHT attribute

NOSMOOTH: Turns off the [smooth mode](#)

ROMREMAP: Sets PDHFIC to use OS3's ROM routines to do the colour mapping.

See [OS3 Mapping](#)

ALTPAL: Alternative palette to map to in ROMREMAP mode. The default palette is the ZX Datatype one.

NOHEADER: Stops PDHFIC from writing .header files when saving .bytes

CLIPBOARD: Sets PDHFIC to read from the clipboard instead of a file. If you specify this option, then DTFILE will specify the clipboard unit number.

NOCFG: Not yet implemented.

Will specify that PDHFIC should not read the default config file.

QUIET: PDHFIC will not display any output except "conversion failed".

To find out why, you may have to run the same conversion again with QUIET switched off.

* Required options

Examples;

PDHFIC ram:pic.png

will convert the ram:pic.png to an SCR called ram:pic.scr

PDHFIC ram:pic.iff ram:specpic SAVEFORMAT ZX82 NOSCALE

will convert ram:pic.iff to a ZX82 file named specpic, and will NOT scale the source image down to the correct size.

PDHFIC 0 ram:clipboard.scr CLIPBOARD

will convert the picture stored in clipboard unit 0 to a SCR called ram:clipboard.scr

PDHFIC 2 CLIPBOARD

will.. er... probably create a file "2.scr" in the current directory, from the contents of clipboard unit 2. It might not though.

1.52 Tooltypes

Available Tooltypes: Note that the tooltypes do not exactly correspond with the CLI arguments.

COLOURSENSE= <value> [default = 20]

BRIGHTSENSE= <value> [default = 450]

WHITESENSE = <value> [default = 200]

These set the colour levels at which point PDHFIC detects colour, white from black and the BRIGHT attribute. Please note that setting any of these to 0 will put them to the default. See [colour mapping](#) for information on how to use these.

GREENMAGENTA= <value> [default = 0]

BLUECYAN = <value> [default = 20]

REDYELLOW = <value> [default = 40]

These set the detection of magenta over green, cyan over blue and yellow over red. The higher the number, the more likely the primary colour is to appear. Use these if you get reds appearing when yellows should (for example)

FLASHRED=<value>

FLASHGRN=<value>

FLASHBLU=<value>

These set when to use the FLASH attribute. See [FLASH mode](#) .

NOSCALE

This turns off the "dynamic re-scaling" feature of PDHFIC.

NOBRIGHT

Turns off the BRIGHT attribute matching (same as BRIGHTSENSE=5000). Please note that if both BRIGHTSENSE and NOBRIGHT are specified, BRIGHTSENSE will do nothing.

GREYSCALE

This stops PDHFIC from mapping colours (same as COLOURSENSE=5000). Please note that if both GREYSCALE and COLOURSENSE are specified, then COLOURSENSE will do nothing.

NOSMOOTH

This turns off the smoothing feature of PDHFIC. See [colour mapping](#)

ROMREMAP

PDHFIC will use OS3's built-in colour mapping routines instead of the custom ones. See [OS3 Mapping](#)

ALTPALETTE

Sets ROMREMAP mode to use an alternative palette when mapping. The default palette is identical to ZX Datatype's one.

SAVEFORMAT = SCR | TAP | ZX82 | BYTES | TZX | GIF

Sets the default format to show on the GUI.

ADVANCED

Enables the [advanced options](#) panel at startup.

SCREEN

If specified, MultiView will be run on its own screen.

CLIPUNIT = <0-255>

Specifies the Clipboard unit number to use. Default is the primary unit (0)

WARNOVERWRITE

If set, PDHFIC will warn you if you have selected an existing file for writing.

TZXBLOCKS = INFO | STANDARD | CUSTOM

This sets which blocks should be written in TZX mode.

INFO: Write an [archive info](#) block

STANDARD: Write a [standard loading block](#)

CUSTOM: Write a standard [custom block](#)

Note that only one of STANDARD or CUSTOM can be specified.

NOHEADER

This will stop the .bytes saver from creating corresponding .header files.

NOPROGRESSBAR

This stops the progress bar window from opening.

NOFILTER

This will turn off the input file requester's intelligent filtering. Use this option if the input file requester does not work.

ODDCOLOURS = INK | PAPER

Select whether any "unmapped" pixels should be INK or PAPER

NOAUTONAMING

Will stop PDHFIC's **auto-naming** feature.

WINX = <value>

WINY = <value>

These will set where PDHFIC's window appears on-screen. Use the **snapshot window** to set these easily.

1.53 The Input File Formats

Datatypes

As you probably already know, this is a feature of Workbench 3.0+ allowing applications to read (potentially) any data type. PDHFIC can read pictures through the Datatypes system.

PPM

This is a Portable Pixel Map (or, more correctly, a "portable pixmap"). PDHFIC supports "P6" type files, which are binary/rawbits PPMs. They *must* be in the correct size (256x192).

Please note that if you have a PPM datatype, this will be used in preference (and then you don't need to ensure your pictures are the right size). The disadvantage is that loading through datatypes will be slower.

1.54 The Output File Formats

SCR (Raw SCREEN\$)

SCR is basically no more than a simple file dump of the Spectrum's screen memory: starting at byte 16384 and lasting for 6912 bytes. The actual format of a Spectrum screen is difficult to explain, but if you've ever watched a Speccy load a screen from tape then you already know the format anyway. If you haven't, then just try it. It's certainly a lot easier than me attempting to explain it to you.

If you want to view SCR files, then the best thing to try is the ZX Datatype.

TAP (Z80 Tape Image)

This is a tape image format used by the emulator Z80 on the PC.

Currently the only emulator on the Amiga that can read these is ZXAM (through an ARexx script).

TZX (ZX Tape)

This is a "digital tape" format, which can store turbo loading blocks as well as a lot of other information.

There is also a custom block defined in the ZX Tape format, which is specially for Spectrum screen files. I have no idea what can read these (if anything), but you can write in this format anyway by toggling the "TZX Blocks » Custom" menu item.

The TZX output is entirely based on the v1.12 documentation - I have NOT been able to test this much, if you can then please let me know whether it works or not!

Currently the only programs on the Amiga that can read these are TZX to TAP conversion utilities...

BYTES (.header/.bytes)

This is the Amiga's standard format for tape files on disk.

ZXAM and Peter McGavin's Spectrum can read these.

ZX82

This is another standard format on the Amiga, which does not quite conform to how a Spectrum would save data (on tape): ie. it has a special header, and the filename comes from the actual name of the file. There is also no checksum. PDHFIC can currently only save the non-compressed files.

Speculator, ZXmit and the ZX Datatype can read this format. There are some other programs which also use it, and details can be found in the Speculator documentation.

To read headered SCR files into an emulator, simply set it to the appropriate load mode and type LOAD "" SCREEN\$ [then press return] This can be achieved in 48K BASIC by pressing the keys; [J] [Symbol Shift and P] [Symbol Shift and P] [Symbol Shift and Caps Shift] [Symbol Shift and K] [Return]

GIF GIF87a

The generally-accepted standard for Web graphics (BOO! Please join my campaign to popularise PNGs and crush GIFs into the ground. Hurrah!). Er.. where was I? Oh yes, PDHFIC can create GIFs for people wanting to use Spectrum SCRs on their web pages, through use of the excellent SCR2GIF [<ftp://ftp.gns.getronics.nl/pub/os/sinclair/tools/amiga/scr2gif.lha>]. If you want to use this feature, please drop SCR2GIF into the same drawer as PDHFIC.

1.55 Future Plans

This is a list of things I might possibly add for the next release (in no particular order):

Saving "into" TAPs (and possibly other tape formats) using the forthcoming, er, hold on... this might be top secret...

Compressed ZX82 files

Decompressing input files

Dither pics down to Spectrum palette. Er... no idea how to do this.

"Everything else is PAPER" option (see colour mapping section)

More presets for the favourite SENSE options (like the current GREYSCALE and NOBRIGHT switches). If I can think of any. ALLBRIGHT comes to mind.

Presets in a menu on the GUI?

Change the BRIGHT detection so it only considers the brightness of INK and PAPER, not every colour in the block. (easy to add for OS3 version...)

Dynamic Contrast for checking the brightness of the picture and selecting the correct level for BRIGHT to make everything look good.

Modular system for savers (add extra output file formats with a few clicks of the mouse. Or a text editor. Or something.)

Warn the user if no FLASH pixels were found, when FLASH is set

Of course, if you have any other suggestions...

1.56 Thanks To...

People who deserve a mention include;

Blood for his BMP2SPEC program, which gave me the inspiration to create this

Gerard Sweeney who was the only person who seemed vaguely interested in an Amiga-specific xxx2SCR type program. Until I mentioned that I was going to write one anyway. At which point everyone seemed interested.

Simon Goodwin who made some suggestions and gave me the information required to implement the ZX82 format.

Paul Hill for telling me about more bugs than anybody else...

And, of course, all those lovely people on comp.sys.sinclair

1.57 Contact Information

This program was written in 1998 by Chris Young and Joe Mackay

(just in case you don't read the text on the contents page)

As always, the best contact method is by e-mail, so that's...

unsatisfactory@bigfoot.com (Chris Young)

joe.moore2@ukonline.co.uk (Joe Mackay)

If you have any suggestions, queries or anything then send it to me (Chris) and I'll listen and most probably reply as well.

If you are having problems with specific datatypes, then send it to Joe as he wrote the original Datatypes handling code. Hehe.

Also check out the Unsatisfactory Software website at;

<http://www.unsatisfactory.freemove.co.uk/>

Note that if you want to directly access subdirectories (eg. the PDHFIC/ppmtoscr support area), then you **MUST** include a trailing slash! If you don't, then it won't work!

Please check the PDHFIC pages for latest bug reports and other information.

1.58 Colour Mapping Algorithm

The colour mapping in PDHFIC has to take a couple of things into account:

1. The Spectrum has a fixed palette consisting of 15 colours (inc BRIGHT)
2. Each 8x8 block of pixels can only contain two colours out of eight.

Basically, if you think the colour mapping in PDHFIC is crap, then you're probably right. However, please read the following if you are having trouble getting something to look good;

PDHFIC's "super" colour mapping takes the two **MOST USED** colours in each block and maps one to PAPER and the other to INK. Then each 8x8 block is checked for pixels of a brightness above the BRIGHTSENSE level (see **tooltypes**). If over half are above this level, then the block will be given the BRIGHT attribute.

It also incorporates a new "smooth" mode (enabled by default) - any pixels not mapping as INK or PAPER in a block will map to the closest colour. This can remove excess pixels and give images a more, er, "smooth" look. What it is actually designed to do, however, is make colour choices more accurate. If disabled with the NOSMOOTH switch/tooltype, any excess pixels will map to INK. If you are getting cases of disappearing pixels, try setting NOSMOOTH.

Pictures with a lot of shading, due to the limited palette on the Spectrum, will normally come up with large areas of the same colour. If this happens, then dither the picture down to two colours. Convert it again, and add the colour back in afterwards. The dithering should make it look a lot better. However, brightly coloured digitised images can convert very well.

Greyscale pictures, can look quite good without needing to be dithered. In fact, colour images saved in GREYSCALE mode can also look quite good, and are worth trying if saving in colour mode didn't yield very good results. Tweaking the WHITESENSE and BRIGHTSENSE options in GREYSCALE mode will change the contrast of the resulting image.

Pictures with large areas of bold colour, and sharp contrast (such as cartoons) will convert about 6 millions times better than real life type images. Try to avoid having a lot of different colours close to each other.

If the picture you are trying to convert is quite dark, try reducing the values of BRIGHTSENSE and WHITESENSE. If you are converting a bright picture, then decrease these values. The BRIGHTSENSE option especially applies to greyscale pictures.

If you often get colours being detected as greys, try reducing the COLOURSENSE value.

Also lower COLOURSENSE (or the individual fine adjustments REDYELLOW etc options) if you are getting reds, greens and blues instead of cyans, magentas and yellows. Raise it if the opposite is happening.

To stop the BRIGHT attribute being used, set BRIGHTSENSE to an unearthly large value (such as 5000), or set the CLI switch "NOBRIGHT"

To only convert pictures into greyscale, set COLOURSENSE to a large value (such as 5000), or set the CLI switch "GREYSCALE"
 To get totally monochrome images, set both BRIGHTSENSE and COLOURSENSE to 5000, or specify both NOBRIGHT and GREYSCALE on the command line.

Note: It is much easier to test these options using the [advanced options](#) part of the GUI.

1.59 Who in their right mind is actually going to use this feature?

PDHFIC includes the FLASH attribute. How it decides what to flash is quite simple, but you have to be quite careful when trying to use it.

The FLASHxxx tooltypes/arguments tell PDHFIC which blocks to flash. When it comes across the colour you specify, the block that colour is in will be given the FLASH attribute.

Warning: Scaling may make the FLASH colour disappear completely! I recommend turning off scaling when using FLASH mode.

To specify the colour to symbolise a FLASH, use the FLASH tooltypes or arguments as follows;

tooltype command line arguments

FLASHRED=<value from 0 to 255> FLASHRED <value> or RED <value>

FLASHGRN=<value from 0 to 255> FLASHGREEN <value> or GRN <value>

FLASHBLU=<value from 0 to 255> FLASHBLUE <value> or BLU <value>

You must specify all three for it to work, and using black (RED=0, GRN=0 and BLU=0) probably won't work.

This should describe the exact colour component (in decimal) of the colour you want to trigger a FLASH. This is the colour on the ORIGINAL PICTURE, not the SCR!

You can also use the GUI's [advanced options](#) - if possible, PDHFIC will show you which colour will flash.

The way I advise you to use this is to set a colour aside for use with FLASH, and assign it a colour close to the one where it is going to be used. That way, PDHFIC won't pick up one pixel of a completely different colour (and therefore possibly mess up the colour mapping)

I'd be very interested if anybody actually puts FLASH to good use - let me know if you use this feature!

(Perhaps some people might like to use it to create crap "animated" SCRs..?)

Oh, one more thing - for obvious(?) reasons, the ZX Datatype doesn't implement the FLASH attribute. And the non-AGA versions of ZXAM also can't display it.

1.60 Example SCRs

The example SCRs are as follows;

mansell.scr (PPMtoSCR v2.0;smooth)

Here we have ol' Nige demonstrating how good colourful digitised images can look on the Speccy. Not bad.

gromit.scr (PPMtoSCR v1.0)

This was a colour image, but has been dithered down to two colours before conversion. This is similar (if not identical) to the output you would get from BMP2SPEC. It looks good, but would look even better when/if re-coloured.

cartoon.scr (PDHFIC v2.0;smooth)

After a lot of fiddling with PDHFIC's [advanced options](#), I managed to get this out of it. Pretty darn close to the original, I'd say.

The Windows Logo

Spookily, this converted very well, but I'm scared of MS and therefore unable to distribute it.

More

Yes, folks. There's lots more examples on the Unsatisfactory Software web pages...

1.61 Program History

PPMtoSCR history removed! If you want to read about the forerunner to PDHFIC, then consult the ppmtoscr archive.

Releases with letters appended to the version number are BETA and should be replaced with the full version as soon as possible.

Version 2.5 (26.12.98)

Just minor fixes, I think. I can't really remember. This version has been sitting around on my HD for a while, and I thought I may as release it.

Version 2.4 (31.10.98)

- * PDHFIC will now warn you if extscale is not available
- * The information requester will now tell you how many FLASH pixels were found.
- * Will transparently use SCR2GIF to create Spectrum screens in GIF format (if such a thing is possible!)
- * Snapshot window option

Version 2.3 (03.10.98)

- * v2.1 crashed on startup if the user didn't have a free pen. Fixed.
- * Added an option to disable the "Intelligent Filtering", as this seemed to be causing a problem on some systems.
- * Fixed the Installer script: it now adds the PPaint config line for SaveSCR at the END
- * The TZX saver now has an option to change the border colour of custom block SCREEN\$. In addition, the description in custom blocks now contains the annotation text from the original file (if available). It also now generates TZX files of v1.12
- * Can switch off the progress window if it annoys you.
- * When smoothing mode is off, you can now select whether you want all the odd pixels mapped to INK or PAPER.
- * Will now automatically decide on an output filename, and you can even just select a directory with no filename (this may have caused problems in previous releases)
- * Changed "warn overwrite" so it warns before writing the file, rather than when selecting it.
- * Fixed a bug with calling extscale and using relative directories.
- * Fixed every known Enforcer hit in the program (including a few that I didn't know about)

Version 2.2 (02.10.98)

Internal testing only - not for general release.

Version 2.1 (27.09.98)

- * Oops - v2.0 wasn't creating valid SCRs! Fixed.
- * Added "Write .header" option to enable/disable writing of .headers with .bytes files.
- * Fixed the SaveSCR.pprx script

Version 2.0 (20.09.98)

- * Made PDHFIC's GUI into an AppWindow
 - * The GUI is now font sensitive
-

- * Added lots of extra options to the GUI, including a new Advanced Options section
- * Added an ARexx port
- * Can now load files from the system clipboard
- * Re-added the PPM support in (sort of) - PDHFIC can now load PPM P6 files, if they are exactly 256x192 pixels in size.
- * There is now a filter on the input file requester, which will only show up valid Datatypes images and PPMs of the correct size. It took me ages to work out how to do that, so you'd better appreciate it. Right?
- * Added TZX support, with a new TZX Blocks menu
- * New Display options for viewing "before and after" pictures
- * More information in the info requester.
- * Fixed the bug causing Enforcer hits on the CLI interface, and also causing strange things to appear on the Speccy's filename.
- * Scaling now works from the CLI interface.

Can anybody tell I've just had a week off work, with nothing better to do than run through a list of the Amiga's ROM functions, implementing each and every one of them into PDHFIC? If someone can cure me from this sadness, please do...

Version 1.0 (28.08.98)

First release.

Enhancements from PPMtoSCR 2.1 include;

Directly supports Datatypes

Small GUI

More colour mapping adjustments

Alternative OS3 palette to use

1.62 Known Bugs (most important first)

If you run any ASL patching utility (RTPatch, ReqChange etc), then you might experience problems with the source file requester. If this happens, DISABLE INTELLIGENT FILTERING. Sorry, but it's some kind of weird side-effect of my ASL filter. Can't you lot just use the (unofficial?) asl.library v42 on Aminet? That should work.

Sometimes when quitting PDHFIC, the window either doesn't close or the Miggy GURUs. Er... if this happens to you, can you send me a quick email describing the steps leading up to it, the current settings etc. Thanks.

Rather annoyingly, the BRIGHTSENSE slider on the GUI doesn't seem to be working properly (or at all, even).

Currently when scaling more than one pic from the GUI, PDHFIC will call the program "extscale". This is just a temporary workaround until I find out what causes the problem. If you want to, you can make your Amiga crash by enabling the tooltype "TESTMODE" and then scaling more than one pic.

The "Use Screen" menu option shows that the option is off when starting PDHFIC, even if it isn't.

1.63 The OS3 ROMREMAPping

ROMREMAP attempts to map pictures to the closest colours in the Spectrum's palette (choice of the ZX Datatype's colour palette or a different one)

In theory, this should give much better results than my current routine. However, it seems to map rather too many colours to white.

Please try converting pictures with this option, and **let me know** how much success you have.

Some pictures do look good using the OS3 routines, but I think I need to adjust the colour map quite a lot.

If you want to convert SCRs into TAPs or whatever, then the ROMREMAP does a better job than the original... (just make sure you aren't using ALTPALETTE)

Limitations

All colour mapping adjustment features are not implemented. eg. GREYSCALE NOBRIGHT, COLOURSENSE, BRIGHTSENSE, WHITESENSE, REDYELLOW etc will do nothing.

It is slower than the original

1.64 The Competition Results

I recently ran a small (sort of) competition on my website, where I asked people to think of a name for this program.

Response? No official entries (unless you count the one I sent to myself to check the form was working)

My site has had a couple of hundred hits in the last month, where the compo was clearly displayed. OK, admittedly most of those people were probably quickly taken in by YSAC and, if not, probably recognised the name "Stuart N Hardy" and took a look at that instead. Before remembering who he is and then quickly veering off on a sharp tangent to some other site. But anyway.

I can't believe you lot. Out of (let's say) 200 hits on the index page, at least a tenth of those must have been on their way to the PPMtoSCR area. That's twenty people. And not one of them could think of a name. Oh well.

So, it was back down to me. DT2SCR was the obvious choice, and has been used even from before I thought about writing it. With a lack of ideas, #?toSCR was a popular candidate, but could prove difficult in the field of CLI usage. It had a longer cousin, featuring several different wildcards and some bad pattern-matching, but I've completely forgotten what it was exactly. So, enter Gerard's poor-quality suggestion of "Supertest Loading Picture" (or something), which apparently equated to DT2SCR (? - don't ask me*) Not good. Then there were my equally pathetic suggestions, including DaToS (Datatype To Spectrum or SCR or something - perhaps that should be DaToSoSoS???) and TAFKAP (The Application Formerly Known As Ppmtoscr). Er... yes.

So, Joe Mackay comes to the rescue again. With PDHFIC. So... why not another competition? The first person to correctly guess what it stands for wins, er, nothing. As I explained last time: I can't afford to give out proper prizes.

* Government Health Warning: Knowing how that one worked will seriously damage your health.

1.65 ARexx Port

PDHFIC has an ARexx port, which you can use to control most aspects of the program. Theoretically, you could write a script to batch-convert images, or to save data from an image processor "through" PDHFIC to create SCRs.

If you create any scripts for PDHFIC, please send them to me for inclusion in the next release.

The port is called "PDHFIC", and here are the commands;

ALTPALETTE

BLUECYAN

BRIGHTSENSE

COLOURSENSE

CONVERT

FLASHBLUE

FLASHGREEN

FLASHRED

GETALPAL

GETBLUCYN
GETBRIGHT
GETCOLOUR
GETDEST
GETFLASHBLU
GETFLASHGRN
GETFLASHRED
GETGRNMAG
GETREDYEL
GETROMREMAP
GETSAVEFORMAT
GETSCALE
GETSMOOTH
GETSOURCE
GETWHITE
GREENMAGENTA
NOSCALE
NOSMOOTH
QUICKCONVERT
REDYELLOW
ROMREMAP
SAVEFORMAT
SCALE
SELECTDEST
SELECTSOURCE
SMOOTH
STDREMAP
VERSION
WHITESENSE
ZXDTPALETTE

Included scripts

1.66 arexx-convert

CONVERT

Will act as if the user clicked on "Convert!" - ie. it will convert the selected files using the current settings.

RESULT: None

RC: 0 - OK

1 - Convert not available (source or dest not selected)

2 - Error during conversion

1.67 arexx-selectsource

SELECTSOURCE [filename]

Displays a file requester for the user to select the source DT image.

If the optional source filename is specified, then PDHFIC will set the source to that image, if possible.

[2.3] If auto-naming is enabled, this will also set the destination to a sensible default. Note: You cannot rely on this being enabled! There is currently no way to check from ARexx whether it is or not.

RESULT: Complete path and filename of selected image.

RC: 0 - OK

1 - Error (No file selected or not valid image)

NOTE: If you supply the optional filename, you must write the command as follows;

Method 1.

```
ADDRESS PDHFIC 'SELECTSOURCE "RAM:mypic.iff"'
```

Method 2.

```
ADDRESS PDHFIC
```

```
SELECTSOURCE '"RAM:mypic.iff"'
```

1.68 arexx-selectdest

SELECTDEST [filename]

Displays a file requester for the user to select the destination filename.

If the optional filename is specified, then PDHFIC will set the destination to that file.

RESULT: Complete path and file selected.

RC: 0 - OK

1 - No file selected

NOTE: If you supply the optional filename, you must write the command as follows;

Method 1.

```
ADDRESS PDHFIC 'SELECTDEST "RAM:mypic.iff"'
```

Method 2.

```
ADDRESS PDHFIC
```

```
SELECTDEST '"RAM:mypic.iff"'
```

1.69 arexx-quickconvert

QUICKCONVERT <source> <dest>

Will convert <source> to <dest> using current settings.

RESULT: None.

RC: 0 - OK

1 - Source file is not a valid image

2 - Error occurred during conversion

NOTE: You must supply the filenames in quotes, which basically means you need two lots of quotes. Er... like this, that is;

Method 1.

```
ADDRESS PDHFIC 'QUICKCONVERT "RAM:mypic.iff" "RAM:mypic.scr"'
```

Method 2.

```
ADDRESS PDHFIC
```

```
QUICKCONVERT '"RAM:mypic.iff" "RAM:mypic.scr"'
```

1.70 arexx-scale

SCALE

Switches scaling mode ON

1.71 arexx-noscale

NOSCALE

Switches scaling mode OFF

1.72 arexx-romremap

ROMREMAP

Switches ROMREMAP mode ON

1.73 arexx-stdremap

STDREMAP

Switches ROMREMAP mode OFF

1.74 arexx-altpalette

ALTPALETTE

Switches to the alternative ROMREMAP palette

1.75 arexx-zxdtpalette

ZXDTPALETTE

Switches to use the ZXDT palette for ROMREMAPping

1.76 arexx-smooth

SMOOTH

Switches smooth mode ON

1.77 arexx-nosmooth

NOSMOOTH

Switches smooth mode OFF

1.78 arexx-brightsense

BRIGHTSENSE <value>

Changes to the new <value> for BRIGHTSENSE

1.79 arexx-coloursense

COLOURSENSE <value>

Changes to the new <value> for COLOURSENSE

1.80 arexx-whitesense

WHITESENSE <value>

Changes to the new <value> for WHITESENSE

1.81 arexx-redyellow

REDYELLOW <value>

Changes to the new <value> for REDYELLOW

1.82 arexx-bluecyan

BLUECYAN <value>

Changes to the new <value> for BLUECYAN

1.83 arexx-greenmagenta

GREENMAGENTA <value>

Changes to the new <value> for GREENMAGENTA

1.84 arexx-flashred

FLASHRED <value>

Changes to the new <value> for FLASHRED

1.85 arexx-flashgreen

FLASHGREEN <value>

Changes to the new <value> for FLASHGREEN

1.86 arexx-flashblue

FLASHBLUE <value>

Changes to the new <value> for FLASHBLUE

1.87 arexx-version

VERSION

Returns the version of PDHFIC that is running.

You should check this value when using new ARexx commands which aren't implemented in older versions.

1.88 arexx-saveformat

SAVEFORMAT <value>

Changes the current **output format** . <value> can be a number from 0-4, which represents;

0 - SCR 1 - ZX82 2 - BYTES 3 - TAP 4 - TZX

1.89 arexx-getsaveformat

GETSAVEFORMAT

Returns the current output format (SCR, ZX82, BYTES, TAP or TZX)

1.90 arexx-getsource

GETSOURCE

Returns the current source path and filename

1.91 arexx-getdest

GETDEST

Returns the current output path and filename

1.92 arexx-getbright

GETBRIGHT

Returns the current BRIGHTSENSE level

1.93 arexx-getcolour

GETCOLOUR

Returns the current COLOURSENSE level

1.94 arexx-getwhite

GETWHITE

Returns the current WHITESENSE level

1.95 arexx-getredyel

GETREDYEL

Returns the current RED/YELLOW level

1.96 arexx-getblucyn

GETBLUCYN

Returns the current BLUE/CYAN level

1.97 arexx-getgrnmag

GETGRNMAG

Returns the current GREEN/MAGENTA level

1.98 arexx-getflashred

GETFLASHRED

Returns the current RED level for FLASH

1.99 arexx-getflashgrn

GETFLASHGRN

Returns the current GREEN level for FLASH

1.100 arexx-getflashblu

GETFLASHBLU

Returns the current BLUE level for FLASH

1.101 arexx-getscale

GETSCALE

RESULT: 0 - Scaling mode is OFF

1 - Scaling mode is ON

1.102 arexx-getsmooth

GETSMOOTH

RESULT: 0 - Smooth mode is OFF

1 - Smooth mode is ON

1.103 arexx-getaltpal

GETALTPAL

RESULT: 0 - Using ZX Datatype palette

1 - Using alternative palette

1.104 arexx-getromremap

GETROMREMAP

RESULT: 0 - Using standard colour mapping

1 - Using OS3's built in ROM routines

1.105 The ARexx Scripts

LoadDT.zxam - This is a script for the emulator ZXAM, which will allow you to load Datatypes images directly into the (emulated) Spectrum's screen memory.

SaveSCR.pprx - This is a script for Personal Paint, which should be installed into PPaint's macro section. It allows you to save pictures from PPaint, and directly convert them into SCRs (or whatever).

1.106 Frequently Asked Questions (which I largely just made up)

Why/what/how the blimmin' 'eck...?

I'm sorry, but if you're going to talk like that then I won't help you.

Are you really as nasty as you make out to be in this guide?

No.

Er... so what the blimmin' eck...

Hey!

I mean, what exactly does PDHFIC stand for then?

I'm not telling you. Keep guessing.

Why does the progress bar not work properly if I try to scale more than one pic?

It's because of my workaround for the bizarre bug in the scaling routine, which the greatest minds in the world can't actually track down. Anyway, it happens to be Joe's fault as he wrote that bit of code.

If you really want to, you can enable the TESTMODE tooltype. This will force PDHFIC to always use the internal scaling routine, but be warned that the second scale you do will almost certainly crash the Amiga. (if it doesn't, then let me know)

So what causes this scaling bug, then?

I have no idea. If I knew, then I would have fixed it.

The input file requester doesn't actually work! What can I do?

Enable the tooltype NOFILTER, or deselect the "Intelligent Filtering" option from the menu. Alternatively, quit any ASL patches you are running (reqchange reportedly crashes the system completely, and RTpatch causes illegal memory accesses). This should make it work.
