ppmtoscr

Chris Young

COLLABORATORS			
	TITLE : ppmtoscr		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Chris Young	August 26, 2022	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

1 ppmtoscr **1**

1.1 Welcome to ppmtoscr... 1
1.2 DISCLAIMER 2
1.3 Introduction to ppmtoscr 2
1.4 Distribution 3
1.5 Requirements 3
1.6 Installation 4
1.7 Usage Instructions 4
1.8 ilbmtoscr 4
1.9 Obtaining a suitable input file 5
1.10 Command Line Arguments 5
1.11 File Formats Information 6
1.12 The Input Formats 7
1.13 The Output File Formats 7
1.14 Future Plans 8
1.15 Thanks To... 9
1.16 Contact Information 9
1.17 Colour Mapping Algoritm 9
1.18 Who in their right mind is actually going to use this feature? 10
1.19 Example SCRs 11
1.20 Program History 12
1.21 Known Bugs 13
1.22 The OS3 ROMREMAP 14

Chapter 1

ppmtoscr

1.1 Welcome to ppmtoscr...

```
                ppmtoscr 3.3 by Chris Young
© 1998 Unsatisfactory Software

*************************************************************************
*If you are expecting this to convert IFF/DT images, with a nice little GUI*
*be warned that you really should be looking at gfx/conv/pdhfic.lha instead*
*************************************************************************


                *DISCLAIMER*
                 Hey!  Can I sue?


                Introduction
                 What is ppmtoscr anyway?

                Distribution
                 Can I give it to my friends?

                Requirements
                 What do I need?

                Installation
                 How do I get it on my HD?


                Instructions
                 How do I use it? *PLEASE READ*

                OS3 Mapping
                 What's so special about ROMREMAP?


                File Formats
                 What are these PPMs and SCRs?

                Col Mapping
                 How does the colour mapping work?  Can I improve the output?
```

FLASH Mode
 Who... I repeat, WHO is going to use that? Eh?

Example SCRs
 What will I get?

Future Plans
 What will the next version do?

Known Bugs
 Which bits don't work?

Prog History
 How crap were the older versions?

Thanks to...
 Who helped or inspired you?

Contact Info
 Who are you anyway?

1.2 DISCLAIMER

ppmtoscr 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 ppmtoscr.

Modified copies of ppmtoscr 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)

ppmtoscr is not in any way based on the source code for Blood's BMP2SPEC program. Neither is it anything to do with the NetPBM distribution, it is simply designed to interface with it.

* 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 Introduction to ppmtoscr

OK. You're probably all wondering what ppmtoscr actually does. In fact, half of you are probably wondering what a PPM actually is. Or a SCR. Or both (in which case, you've left me wondering why you downloaded this in the first place).

ppmtoscr quite obviously converts a PPM to a SCR. It does more than just that, though - it can put header information on the SCR file, to make it into a .bytes, .tap, ZX82 or even TZX instead.

Please note that ppmtoscr 3.0+ is pretty stupid compared to the old PPMtoSCR!

This NEEDS NetPBM. It can't scale or truncate images - this is now down to YOU to make sure pnmscale is called!

A rather complicated story should go here, about what happened to the (pseudo)GUI...

Basically, if you want a GUI and you want to convert Datatypes images directly to SCRs (or whatever) then PDHFIC is the program you want.

1.4 Distribution

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

You must distribute this documentation with ppmtoscr. 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 ppmtoscr through FTP. Everyone else is allowed to do that as well, providing the file is accessible by *everybody* at *no cost* (ie. Anonymous FTP)

Distribution on CD-ROM is perfectly OK. If you are not mentioned below and are thinking of distributing ppmtoscr 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 ppmtoscr 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

1.5 Requirements

ppmtoscr uses several features of OS2.0, so you will need an Amiga running
at least Workbench 2.0

I have not been able to test this program on anything below WB3.0

Recommended are the following;

NetPBM, or more accurately, gfx/pbm/ppmbin2.lha and pnmscale (in
gfx/pbm/pnmbin.lha, I think)

1.6 Installation

Tch. Do you really need to read this? It's OBVIOUS. Just double-click on
"Install ppmtoscr". But remember, kiddies, you need
Commodore/Escom/Amiga In(c|t)'s rather groovy Installer program.

If you don't have it, well, why not?

1.7 Usage Instructions

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

Getting a suitable input file

Command line arguments

The included script - ilbmtoscr
WARNING! ppmtoscr does not check to see if output files already exist -
they will simply be overwritten.

1.8 ilbmtoscr

I have written a (very) short script called ilbmtoscr which may help people
wanting to create SCRs (or whatever) straight away.

Firstly, the programs ilbmtoppm, pnmscale and (of course) ppmtoscr
must be in your current command search path.

Simply run ilbmtoscr from the command line - it takes the following
arguments (all REQUIRED);

ILBMFILE - this is your source IFF ILBM format file

SCRFILE - the filename you want the resulting file to have

SAVEFORMAT - the format of the resulting file (SCR, ZX82, TAP, TZX or BYTES)

(see
output formats
)

Example;

ilbmtoscr RAM:picture.iff RAM:specpic.tzx TZX

will convert (and scale) RAM:picture.iff to the TZX file RAM:specpic.tzx

Please note that ilbmtoscr creates two temporary files called
T:ilbmtoscr.tmp and T:scaledpic.tmp. These will be deleted by the script,
and any other files of those names will be overwritten. Unlikely, I know -
but I'm warning you just in case.

1.9 Obtaining a suitable input file

Find or draw a picture. Pictures without lots of shading will ↩
generally
convert better. If the one you want to convert has got a lot of shading,
then dither it down to two colours, and add the colour back in afterwards
(in Spectrum-style blocks). Saving in
greyscale
mode can also
often improve the resulting image quality. However, bright and colourful
digitised images can look very good in colour.

It is advised that the source image should have an aspect ratio of 4:3 or
similar (even better if you can cut or scale the picture down to 256x192)

Assuming you can't save directly in PPM format, you will then need to run
the picture through a xxxtoppm conversion utility. These are available on
Aminet in gfx/pbm/ppmbin2.lha. PPaint's PBM saver is perfectly capable of
saving PPMs, but please save with the BINARY mode ON, and AUTOMATIC switched
OFF - else you are likely to end up with a file that ppmtoscr can't read.

The source image MUST be of size 256x192 before being run through
ppmtoscr. Call pnmscale with the following;

pnmscale >ram:scaledpic.ppm -xsize 256 -ysize 192 ram:pic.ppm

Where ram:pic.ppm will be the source image, and ram:scaledpic.ppm is the one
you need to feed into ppmtoscr.

The PPM file must be a P6 (PPM rawbits/binary) format file.

1.10 Command Line Arguments

Shell Template:

```
PPMFILE/A,SCRFILE/A,FORM=SAVEFORMAT/K,COL=COLOURSENSE/K/N,
BRT=BRIGHTSENSE/K/N,WHT=WHITESENSE/K/N,GRNMAG=GREENMAGENTA/K/N,
BLUCYN=BLUECYAN/K/N,REDYEL=REDYELLOW/K/N,RED=FLASHRED/K/N,
GRN=FLASHGREEN/K/N,BLU=FLASHBLUE/K/N,GREYSCALE/S,NOBRIGHT/S,NOSMOOTH/S,
OS=ROMREMAP/S,ALTPAL=ALTROMPALETTE/S,NOHEADER/S,QUIET/S

PPMFILE:*    This is your source image file

SCRFILE:*   This is the path and filename of the resulting file.

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=BLUECYAN/K/N,REDYEL=REDYELLOW/K/N

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
                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:  Will stop ".header" files being generated when saving .bytes

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


Example;

PDHFIC ram:pic.ppm ram:specpic SAVEFORMAT ZX82

will convert ram:pic.iff to a ZX82 file named specpic
```

1.11 File Formats Information

```
Input Formats
 PPM

Output Formats
 SCR, .bytes, TAP, TZX and ZX82
```

1.12 The Input Formats

PPM
PPM is a Portable Pixel Map. It is used by the NetPBM distribution programs to convert between various incompatible formats. I choose to use this as the source format because (a) it looked easy to implement and (b) there is a host of programs in Aminet gfx/pbm to convert files to this format*. It's the next best thing to Datatypes.

ppmtoscr can only load "P6" format files: these are PPMs saved with the "rawbits" option. Please note that greyscale pictures saved with PPaint's PBM saver will not save out in the correct format (unless "automatic" mode is switched off)

* Well, that's the whole point.

1.13 The Output File Formats

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

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. It is capable of storing different data blocks (eg. turbo loaders) and other information.

The TZX format has a custom block specially for Spectrum screens, but I have not used that as I have no idea whether it is possible to load up these custom blocks as if they are ordinary Spectrum files.

The implementation of TZX in this program is entirely from the TZX format documentation v1.11 - I have not been able to test it, as there aren't any

programs on the Amiga which actually use TZX yet.

BYTES (.header/.bytes)
This is the Amiga's standard format for tape files on disk.

Most, if not all, Amiga Spectrum emulators can read this format. I don't think any other platform has an emulator capable of reading them.

ZX82
This is another standard format on the Amiga, which does not quite conform to how a Spectrum would save data: ie. it has a special header, and the filename comes from the actual name of the file. There is also no checksum. ppmtoscr 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]

1.14 Future Plans

This is a list of things I might possibly add for the next release:

Change the command line support so it sends the resulting file to stdout in a horrible UNIX fashion :-)

Greatly improve the OS3 colour mapping version

Support other PNM formats (PBM, PGM and the ascii PPM)

Implementing an environmental variable for the default output format, colour modes etc. (I had this implemented for about 10 minutes, but I removed it)

"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.

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.

Saving direct to cassette tape (!)

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

1.15 Thanks To...

A BIG thankyou goes to Joe Mackay, who gave me so much help in writing this that he may as well have written it himself :-)

People who also 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. (Oh, and thanks for pointing out the bug in the installation script!)

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

1.16 Contact Information

This program was written in 1998 by Chris Young.
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

If you have any suggestions, queries or anything then I'll listen and most probably reply as well.

Also check out the Unsatisfactory Software website at;
http://www.unsatisfactory.freeserve.co.uk/

(or http://www.bigfoot.com/~unsatisfactory/ if I suddenly move it for no apparent reason)

Which now also has a ppmtoscr support area. Latest known bugs and bits missing from the documentation will be put up on there for all to see. If possible, please check that page before submitting bug reports!

Please note that this program is linked to PDHFIC, and anything which gets implemented into one program is very likely to end up in the other as well.

1.17 Colour Mapping Algoritm

The colour mapping in ppmtoscr 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 ppmtoscr is crap, then you're

probably right. However, please read the following if you are having trouble getting something to look good;

ppmtoscr's new "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 wield 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 if you are getting reds, greens and blues instead of cyans, magentas and yellows. Raise it if the opposite is happening. (you can also use the new REDYELLOW etc options)

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.

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

As from version 2.1, ppmtoscr 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 ppmtoscr 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: pnmscale does colour averaging when re-sizing images, so this might make some pixels go into your FLASH colour, or make the FLASH colour disappear completely! I recommend making sure your pictures are in 256x192 format when using FLASH.

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

FLASHRED <value> or RED <value>
FLASHGREEN <value> or GRN <value>
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!

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, ppmtoscr 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.

1.19 Example SCRs

The example SCRs are as follows;

mansell.scr (v2.0;smooth)
Here we have ol' Nige demonstrating how good colourful digitised images can look on the Speccy. Not bad.

gromit.scr (Joe Mackay, 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 needs re-colouring.

```
cartoon.scr (v2.0;smooth)
This is a conversion of a cartoon.  You can see that generally the picture
is what you would expect from a Spectrum (and thus is typical of the output
from ppmtoscr).  If you compare this picture with the same one in the 1.0
distribution, you will see how much the colour mapping has improved.

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...
```

1.20 Program History

```
Version 3.3 (26.12.98)
----------

Added the NOHEADER switch

No other changes; I've had this sitting around on my HD for the last couple
of months and thought I may as well release this slight update.


Version 3.2 (10.09.98)
----------

Fixed the bug in the installation script, and a potential problem with the
TZX saver.

Distribution now includes an example "ilbmtoscr" script.


Version 3.1 (04.09.98)
----------

Fixed a bug which may have caused the Spectrum filename to come up as
garbage in TAPs etc.  Caused by me stupidly removing a piece of "redundant"
code.

Added the TZX output format, untested.
  PLEASE - if you can test this, then LET ME KNOW WHETHER IT WORKS OR NOT!


Version 3.0 (28.08.98)
----------

New REDYELLOW etc options for colour mapping tweaking
Removed all the Workbench support*
Removed the automatic calling of ilbmtoppm, pnmscale etc*

* if you want these, then please use PDHFIC instead (gfx/conv/pdhfic.lha)
  ppmtoscr is designed simply for NetPBM usage - its original intention!
```

```
Version 2.1 (29.07.98)
----------

Added support for FLASH (dunno who would want to use it though...)
Added status bar window type thing
Improved the colour mapping still further
Fixed the bug with some tooltypes not working
and if there was a bug with the BRIGHT being permanently on, then, well,
 that's gone now as well...
Proper error messages at last! (ie. it doesn't just come up "Conversion
 Failed!" all the time, it just makes (shock!) sense.  Sometimes.)
SENSE options now available from the Shell
File format selection from the Shell has changed.


Version 2.0 (22.07.98)
----------

Added support for the BRIGHT attribute
Added Shell interface
Added Workbench tooltypes
.tap and .header files now show a proper filename on the Spectrum
Greatly improved "super" colour mapping (configurable)
Includes new "smooth" mode (on by default)
Can now automatically create greyscale and monochrome images
Dynamic rescaling (and everyone thought I was joking...)
Transparently handle IFFs through ilbmtoppm, and DTs through XtoILBM
Removed the filename extension output format detection, and replaced it with
 a requester/command line switches.
New OS3 colour mapping version (preview)


Version 1.0SE (04.07.98)
-------------

Added ZX82 saver
Tweaked the colour mapping algorithm a bit


Version 1.0 (24.06.98)
-----------

First release
```

1.21 Known Bugs

```
None.  Hurrah!


If you find any, then be sure to let me know...

Oh, but I'd appreciate it if you check the Unsatisfactory Software website
to see if I do know about them before submitting bug reports.  Thanks!
```

1.22 The OS3 ROMREMAP

The ROMREMAP option uses the colour mapping routines built in to OS3.

ROMREMAP attempts to map pictures to the closest colours in the Spectrum's palette (choice of ZXDT 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 version, 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 OS3 version does a better job than the original...

Limitations

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

It is slower than the original