

WarpPSD

Oliver Roberts

COLLABORATORS

	<i>TITLE :</i> WarpPSD		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Oliver Roberts	July 31, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	WarpPSD	1
1.1	WarpPSD.datatype 44.2	1
1.2	Description	1
1.3	Features	2
1.4	System Requirements	2
1.5	Installation	3
1.6	Preferences Options	3
1.7	Speed	4
1.8	Distribution Conditions	5
1.9	Disclaimer	6
1.10	Acknowledgements	6
1.11	Support & the Future	6
1.12	About the author	7
1.13	Program History	7

Chapter 1

WarpPSD

1.1 WarpPSD.datatype 44.2

WarpPSD.datatype 44.2 - a 24-bit capable datatype for Adobe Photoshop® images for 68k, PPC/WarpOS and PPC/MorphOS

Copyright © 2001 Oliver Roberts, All Rights Reserved.

Description	what is this datatype for?
Features	list of features
System requirements	what you need to use this software
Installation	installing this software
Preferences	descriptions of configurable settings
Speed	information regarding speed issues
Distribution	distribution conditions
Disclaimer	important notices
Acknowledgements	thankyous and credits
Support & the Future	support + improvements I intend to make
About the author	how to contact the author
History	program history

1.2 Description

WarpPSD.datatype is a datatype which allows you to view and read Adobe Photoshop® image files (commonly used file extensions are .PSD and .PDD). As far as I know, this is the first datatype of it's kind for AmigaOS.

In addition to supporting the 68k (with optimized versions for 020, 030, 040 and 060), it also supports the PowerPC processor, with native WarpOS and MorphOS versions. Even better, it is fast, compact, clean and well behaved - a true plug'n'play PPC datatype. One of the key features is its superior speed, hence the name WarpPSD. This datatype uses the same engine as used by my other WarpDTs.

1.3 Features

- Handles Adobe Photoshop® 2.5, 3.0, 4.0, 5.0 and 6.0 images (currently does not support layered images)
- Most of the available modes are supported, including 24-bit RGB, Greyscale, Monochrome, 256 colour palette-based and CMYK images. Images using RLE compression are also supported
- Highly optimized datatype dispatch engine and PSD decoder, resulting in a very efficient, compact and quick PSD datatype
- Asynchronous file i/o and double buffering techniques (WarpOS only), which speeds up image decoding
- Optimized versions for 68020, 030, 040 and 060
- PowerPC support with native WarpOS and native MorphOS versions
- Alter the pen allocation precision when images are remapped to an 8-bit display
- Specific support for the OS 3.5/3.9 picture.datatype, when available
- The dithering feature of the OS 3.5/3.9 picture.datatype can be configured to your liking (e.g. disabled for 15/16-bit displays)

1.4 System Requirements

This datatype needs the following in order to work:

- Kickstart 3.0 or higher
- picture.datatype v43 or higher (i.e. either of the ones supplied with AmigaOS 3.5/3.9, P96 or CGraphX)

68k version

- 68020 processor or higher (optimized versions included)

WarpOS version

- PPC accelerator card + 68040/060
- WarpUp Release 4 (powerpc.library V15) or higher

MorphOS version

- PPC accelerator card
- MorphOS beta release 1 or higher

Note that a graphics card is not necessary - the P96 or OS 3.5/3.9 picture.datatype will automatically dither images down to your native Amiga display requirements.

1.5 Installation

To install WarpPSD.datatype, simply run the provided installer script by double-clicking the icon. This will create a CPU specific version of the library for you, and install it to SYS:Classes/Datatypes and also installs the PSD descriptor file to DEVS:Datatypes. It does nothing else, so you needn't worry about it messing about with your system :)

1.6 Preferences Options

WarpPSD can be configured using the "Datatypes/WarpPSD.prefs" environment variable. The preferred way of altering the settings is via the WarpDTPrefs graphical user interface (requires OS3.5 or higher), available from <http://www.nanunanu.org/~oliver/warpdtprefs.html> or Aminet (<http://www Aminet.org/ftp/graphics/other/psd/psd.datatype/WarpDTPrefs.lha>).

However, it is also possible to alter the settings manually by changing the environment variable with setenv, according to the following template:

```
DITHER_OVERRIDE/K,DITHER_QUALITY/K,DITHER_DEPTH/N/K,PENS_OVERRIDE/K,  
PENS_QUALITY/K
```

OS3.5 DITHER CONTROL OPTIONS

The three dither options all interact with each other and apply only when at least picture.datatype V44 (OS3.5/3.9) is in use.

By default picture.datatype V44+ dithers images for all target displays less than 24-bit. It is debatable whether it's worth using dithering on 15/16-bit displays - for photographic images, the difference is usually not noticable, but for computer generated images containing smooth colour gradients dithering produces a noticably better output image.

DITHER_OVERRIDE

Selects what our dither quality setting should override:

NOTHING	- do not change the dither quality in any circumstances
DEFAULTS	- only override picture.datatype's default dither quality, which is used when the application does not specify it's own dither quality value
APPS	- override the dither quality that an application may set
EVERYTHING	- always use your custom dithering settings

DITHER_QUALITY

Select the dither quality to use when this datatype decides that changes to the dither quality are required, according to your dither override and depth settings:

POOR	- simple colour remapping and no dithering (fastest)
GOOD	- slower, higher quality dithered output (default)
BEST	- similar to GOOD, with maybe slightly higher quality output

DITHER_DEPTH

Use this option to select which display depths should use the above quality setting. When the target display is deeper than the given depth value, the above dither quality will be applied to that image. For example setting this to 8 will cause the dithering settings to be used for 15/16-bit target displays, but left untouched when the target display is \leq 8-bit.

PEN SELECTION OPTIONS

The pen selection options are only relevant to 8-bit displays, and adjust the precision to which pens/colours are allocated, which affects the image quality and number of pens that will get used.

PENS_OVERRIDE

Selects what our pen/colour quality setting should override:

NOTHING	- do not change the pen precision under any circumstances
DEFAULTS	- only override picture.datatype's default pen precision quality, which is used when the application does not specify
	it's own pen precision value
APPS	- only override the pen precision that an application has set
EVERYTHING	- always use your pen precision settings

PENS_QUALITY

Changes the pen precision quality to use when remapping an image to an 8-bit display, in situations according to your pens otherride setting:

POOR	- use only a small amount of unique colours, at the expense of image quality
GOOD	- allocate enough pens to ensure a reasonable output image quality, whilst not hogging all pens for a single image (default)
BEST	- allocate as many pens as possible/necessary, resulting in the best image quality, but when displaying multiple images on the same screen, the quality of every image might not be so good

1.7 Speed

WarpOS version too slow!

The main problem is that PPC datatypes still have to use the 68k for reading the data from disk and for creating / writing to the bitmap structures that the datatypes system requires. As far as the former goes, time lost for file i/o is negligible as WarpPSD uses double buffered asynchronous i/o (supports DMA controllers).

The largest bottleneck is that the DTM_WRITEPIXELARRAY method of the picture.datatype has to be used to write the image data from WarpPSD into the image bitmap. As this process is done via picture.datatype, it can only currently be performed by the 68k. To give you some idea of how much of a problem this is for WarpPSD, typically, half of the overall decode

time is used by the PPC to decode the whole image, and the other half is used by DTM_WRITEPIXELARRAY on the 68k. And that's on a graphics card - the time used by DTM_WRITEPIXELARRAY will probably be even greater on systems using native Amiga graphics. It doesn't take a genius to see that this is slowing the datatype down, and is the main reason why WarpPSD will still be faster on a 060 than a 040.

How to make the datatype faster

Is there anything that can be done about this? Well, yes, there are a few patches that you can install which should make things faster:

- NewWPA8 (util/boot/NewWPA8.lha on Aminet) should provide a notable speed increase on native Amiga graphics - probably won't make any difference if you use a graphics card.
- If you use a graphics card and CyberGraphX, you may want to make sure you are using the supplied v43 picture.datatype, as this will be faster than the P96 and OS3.5/3.9 picture.datatype on your system.

Of course, any other general speed-up patches should help too.

1.8 Distribution Conditions

WarpPSD.datatype is public domain with the copyright remaining with the author and may be freely distributed legally providing:

- (1) None of the distributed files are changed in any way
- (2) It is not sold for profit and it is not included on any disks that are sold solely for profit (includes magazine coverdisks)
- (3) The distribution contents remain complete (see list below)

If this software is to be sold for profit, permission must be obtained from me, the author.

Aminet and Amigactive have been granted permission to distribute WarpPSD.datatype on their CDs.

The following files must be present in their original and unchanged form in any copies of this software:

```
Classes/Datatypes/WarpPSD.datatype.020
Classes/Datatypes/WarpPSD.datatype.030.pch
Classes/Datatypes/WarpPSD.datatype.040.pch
Classes/Datatypes/WarpPSD.datatype.060.pch
Classes/Datatypes/WarpPSD.datatype.wos
Classes/Datatypes/WarpPSD.datatype.elf
Devs/Datatypes/PSD
Devs/Datatypes/PSD.info
WarpPSD.guide
WarpPSD.guide.info
Install_WarpPSD
Install_WarpPSD.info
spatch
```


1.9 Disclaimer

This software is provided "as is", without warranty of any kind, either expressed or implied, statutory or otherwise. By using the archive and its contents, you accept the entire risk as to its quality and performance.

Neither Oliver Roberts nor any other party involved in the creation, production or delivery of the archive and its contents shall be liable for any direct, indirect, special, consequential or incidental damages, including without limitation damages for loss of profits, loss of use or loss of anticipated costs, expenses or damages, and any data or information which may be lost or rendered inaccurate, even if Oliver Roberts is advised of the possibility of such damages.

Do not attempt to tamper with the supplied files. Doing so will cause problems and you may find things start going wrong!

1.10 Acknowledgements

The PSD decoder routines were written from scratch in order to suit my WarpDT engine, providing maximum performance. Adobe file format specification documents were used as a reference - Copyright © 1991-2000 Adobe Systems Incorporated. Photoshop is a registered trademark of Adobe Systems Incorporated.

Thanks to Nikola Tomic for beta testing and giving me the initial idea of creating a PSD datatype.

The WarpOS version was made possible by VBCC, which was used to build and compile the datatype. Thanks to Volker Barthelmann and Frank Wille, for their support and help.

The Spanish installer translation is by Dámaso D. Estévez.
The French installer translation is by Philippe Bovier.

Thanks also to Sam Jordan for WarpOS and helping me out with various queries regarding it.

Finally, thanks to the OS 3.5 development team - now everyone has access to a 24-bit picture.datatype, I don't need to bother messing about adding dithering routines :)

1.11 Support & the Future

Some things that may appear in the future:

- I plan to add at least some basic support for layers, especially for images that come with a single layer and no background.
- If I can squeeze any more speed out of the datatype, in general, I'll do so :)

If you have any other suggestions, please let me know.

Future releases of WarpPSD.datatype will be available from either Aminet (util/dtype/WarpPSDdt.lha) or its webpage:

<http://www.nanunanu.org/~oliver/warppsd.html>

If you would you like to know when WarpPSD is next updated, then you may want to subscribe to my announcement list to receive an e-mail informing you of the changes as soon as a new versions of any of my products are released. To subscribe, send a blank e-mail to

futaura-announce-subscribe@yahoogroups.com

or go to

<http://groups.yahoo.com/subscribe/futaura-announce>

1.12 About the author

If you have any problems with this software, or if you have any suggestions/queries, please contact me and I will do my best to sort any bugs out as soon as possible:

e-mail: oliver@futaura.co.uk
www: <http://www.nanunanu.org/~oliver/>
icq: 34640231

1.13 Program History

44.2 (24.9.2001)

- Added options to control the precision/quality of pen allocations for when an image is displayed on a palette-based display (8-bit).
- WarpOS version reworked, now working in a fundamentally different way, to completely banish any possibility of cache conflicts.
- Fixed a bug which could cause a system deadlock if the dither control or pen precision options were enabled when using picture.datatype V44 or higher.
- Discovered a bug in the CyberGraphX picture.datatype, concerning error numbers which would have resulted in an unknown error being reported by application software if bitmap allocation failed - added workaround.
- Optimizations made to the dispatch engine.
- PowerPC versions recompiled with VBCC 0.8.
- Added French and Spanish strings to the installer.

44.1 (21.3.2001)

- Initial release.
-