WarpBMP

Oliver Roberts

COLLABORATORS

	TITLE : WarpBMP	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Oliver Roberts	July 31, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

WarpBMP

1.1 WarpBMP.datatype 44.5

```
WarpBMP.datatype 44.5 - a fast 24-bit capable BMP picture datatype
                        for 68k, PPC/WarpOS and PPC/MorphOS
```


1.2 Description

WarpBMP.datatype is a datatype which allows you to view and read Windows or OS/2 BMP image files. A unique feature of this particular BMP datatype is that 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 WarpBMP. This datatype uses the same engine as used by my other WarpDTs.

1.3 Features

- Supports common BMP file types, including both Windows and OS/2 BMP
  formats. These include BMP images with a bit depth of 1, 4, 8, 16, 24
  and 32, both uncompressed and RLE compressed

- Highly optimized datatype dispatch engine and BMP decoder,
  resulting in a very efficient, compact and quick BMP 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 WarpBMP.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 BMP 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

WarpBMP can be configured using the "Datatypes/WarpBMP.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 (util/dtype/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 <= 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

A fast BMP datatype?

The table below shows the time (in seconds) it took to decode 8 different images on my A1200 603e/240MHz 060/50MHz, with BVision and CGX picture.datatype, with other tested datatypes configured as close as possible to WarpBMP's internal settings. However, due to the nature of BMP files (mostly uncompressed, and even the RLE compression used in compressed files is not all that good, and needs nowhere near as much CPU power as JPEG and PNG, for example), don't expect miracles from the PPC versions.

Really, there wasn't much to benchmark my datatype against, apart from Gunther Nikl's BMP datatype. And, as you can see, there's really not a lot in it in terms of speed. The main points to note is that WarpBMP is much faster at decoding compressed BMPs, with the PPC versions also being faster than the 68K version. With uncompressed images, the WarpOS version is around about the same speed as the 68K version, which is not really that surprising. Notabably, the MorphOS version won all speed tests, except for

BMP files that are small in size.

	PowerPC			68K	
	WarpBMP 44.5			G.Nikl BMPdt 40.15	C= v42 PicDT 39.4
	WarpOS	MorphOS	68K		
578x800 24-bit, 1388854 bytes	0.53	0.46	0.52	0.53	n/s
450x366 24-bit, 494886 bytes	0.22	0.20	0.21	0.20	n/s
630x422 24-bit c,794004 bytes	0.33	0.29	0.33	n/s	n/s
320x240 16-bit, 153654 bytes	0.13	0.13	0.32	0.29	n/s
914x891 8-bit c, 573662 bytes	0.28	0.28	0.42	0.67	n/s
640x480 8-bit c, 308246 bytes	0.16	0.16	0.18	0.32	n/s
800x600 8-bit, 481080 bytes	0.22	0.19	0.19	0.20	0.48
320x188 4-bit, 30198 bytes	0.07	0.06	0.05	0.05	0.05

(c = compressed, n/s = not supported)

Note: all tests were performed, multiple times, using Visage, with the following command line: "visage test.bmp time test".

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 negligable as WarpBMP 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 WarpBMP 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 WarpBMP, 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 WarpBMP 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

WarpBMP.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 WarpBMP.datatype on their CDs.

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

Classes/Datatypes/WarpBMP.datatype.020
Classes/Datatypes/WarpBMP.datatype.030.pch
Classes/Datatypes/WarpBMP.datatype.040.pch
Classes/Datatypes/WarpBMP.datatype.060.pch
Classes/Datatypes/WarpBMP.datatype.wos
Classes/Datatypes/WarpBMP.datatype.elf
Devs/Datatypes/BMP
Devs/Datatypes/BMP.info
WarpBMP.guide
WarpBMP.guide.info
Install_WarpBMP
Install_WarpBMP.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 BMP decoder routines were written from scratch in order to suit my WarpDT engine, providing maximum performance. BMP specification documents from various websites, John Bradley's XV and Gunther Nikl's BMPdt and were used as a reference.

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:

- 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 WarpBMP.datatype will be available from either Aminet (util/dtype/WarpBMPdt.lha) or its webpage:

http://www.nanunanu.org/~oliver/warpbmp.html

If you would you like to know when WarpBMP 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

44.5 (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.

44.4 (15.3.2001)

- Added preferences options to configure the OS3.5/3.9 dithering control feature.

44.3 (19.2.2001)

- BMP descriptor no longer insists on BMP files having a .bmp filename extension
- The dithering disabler feature (for when the OS3.5/3.9 picture.datatype is in use on hi/true colour displays) was flawed, since it only worked with images that were eventually attached to windows (e.g. Multiview). Now dithering will be disabled in all other cases too (e.g. Workbench / DOpus backdrops)

44.2 (21.1.2001)

- Speeded up decoding of RLE compressed images (10% faster for PPC)
- Fixed broken handling of RLE compressed 24-bit BMP files

44.1 (17.1.2001)

- Initial release.