

# **BMPdt**

Gunther Nikl

<b>COLLABORATORS</b>
----------------------

	<i>TITLE :</i> BMPdt		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Gunther Nikl	July 8, 2025	

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>BMPdt</b>	<b>1</b>
1.1	BMP picture datatype . . . . .	1
1.2	copyright . . . . .	1
1.3	disclaimer . . . . .	1
1.4	introduction . . . . .	2
1.5	features . . . . .	2
1.6	installation . . . . .	2
1.7	preferences . . . . .	3
1.8	faq . . . . .	3
1.9	acknowledgments . . . . .	4
1.10	history . . . . .	4
1.11	author . . . . .	5

# Chapter 1

## BMPdt

### 1.1 BMP picture datatype

BMP Picture DataType for Workbench 3.0 or above  
Written by Gunther Nikl in 1995/98

FreeWare

Disclaimer  
Introduction  
Features  
Installation  
Preferences  
Datatype FAQ  
Acknowledgments  
History  
Author

Final note: Use at your own risk!

### 1.2 copyright

This BMP package is Copyright © 1995/98 by Gunther Nikl. This software package may be used freely for non-commercial purposes. Distribution of this software package is allowed as long as it remains unaltered.

Hereby permission is granted to distribute this software package on the "Meeting Pearls" CD-ROM series. Distribution on other CD-ROMs, disks series or cover disks requires a permission of the Author.

### 1.3 disclaimer

---

The author takes no responsibility for any results of the use of this program. This software is provided "AS IS" and there is no warranty of any kind, so that you use this software at your own risk.

## 1.4 introduction

Starting with OS Release 3 the Amiga has the concept of 'datatypes', which allow reading and viewing files of different types and formats. MultiView is an application that utilizes these datatypes and handles any file for which you have a data types class installed.

This datatype was created to supersede the bmp.datatype 39.4 from the 3.1 NDK or pictdt\_42\_1.lha. That datatype was limited to 16 and 256 colour pictures, handled only Windows type BMPs (but didn't check for this) and had other quite serious bugs.

Another reason to write this datatype was that I wanted to know whether it's indeed possible to use GNU CC to program one :^)

## 1.5 features

The datatype supports 1, 4, 8 and 24 bit BMPs. It handles OS/2 1.x, Windows 3.x and OS/2 2.x types of BMPs correctly. Images of 4 or 8 bit depth may be rle compressed (it even tolerates possibly corrupted rle images).

Other features:

- switches to V43 mode if a new picture.datatype V43 is found
- asynchronous file I/O to speedup image loading and decoding
- utilizes WritePixelLine8() for chunky-to-planar in V42 mode
- grayscales a true-colour image when working in V42 mode
- can convert a true-colour image into HAM6/HAM8
- can save BMPs (encoder does handle HAM and EHB pictures)
- can be configured to suit personal preferences

Please note:

- 1.) Some types of BMP pictures are called to be in 'DIB' format, but these are simple BMPs only with another extension
- 2.) Pictures from OS/2 Warp are in OS/2 2.x format, thus can be handled by this datatype
- 3.) Compressed BMPs have often a '.rle' file extension
- 4.) Windows9x type BMPs are (currently?) not supported since I don't have any information about this format.

## 1.6 installation

This "BMP" datatype distribution should consist of the following files:

- Classes/DataTypes/680x0/bmp.datatype

- Devs/DataTypes/Windows BMP
- Devs/DataTypes/Windows BMP.info
- BMPdt.guide
- BMPdt.guide.info
- Source code

Copy one "bmp.datatype" into the "SYS:Classes/DataTypes" drawer. The file "Windows BMP" and its info file should be placed in the "DEVS:DataTypes" drawer. In order to use the datatype doubleclick on "Windows BMP.info" (or reboot the machine).

## 1.7 preferences

The datatype can be configured by placing desired options in an environment variable. The name of this variable is "classes/datatypes/bmp.prefs". Please use Setenv (or Echo if you prefer) to create \*global\* or use Set to create \*local\* preference settings. Local settings do override global ones.

Note: You have to copy the variable from ENV: to ENVARC: if you want to keep global settings permanently!

The preference template is:

```
V42MODE/S,GRAY/S,HAM/K/N
```

```
V42MODE
```

```
-----
```

Forces the datatype to work in the old V42 mode even if a new V43 picture datatype is installed. The datatype switches automatically into this mode if no V43 aware picture.datatype is found.

```
GRAY
```

```
----
```

If specified the datatype emits grayscale images.

```
HAM
```

```
---
```

Request the conversion of true-colour data into HAM. A value of 6 invokes HAM6 and a value of 8 invokes HAM8 mode.

Note: This option is ignored if grayscale is enabled or if the supplied argument is invalid.

## 1.8 faq

Q: Why is the datatype so slow with the P96 picture.datatype V43?

A: The picture.datatype coming with P96 dithers 24bit data even for 15/16 bit screens. Either use only 24bit screens or disable dithering globally this way (posted on the PML by Olaf "Olsen" Barthel):

```
setenv classes/datatypes/picture/dither 2
```

and then following it up with

```
copy env:classes envarc:classes all
```

Note: it may be necessary to create the picture drawer manually to get the Setenv command succeed!

Q: Can I speedup decoding in V42 mode?

A: Since the OS function `graphics.library/WritePixelLine8()` utilized for c2p conversation when working in V42 mode is not the fastest one possible you should consider installing a patch for this function. Highly recommended is NewWP8 from Michael van Elst available on AmiNet.

Note: This probably affects native graphic chip-sets only!

## 1.9 acknowledgments

This "BMP" datatype was written from scratch using GNU CC 2.7.2.1 and EGCS 1.0.3a (ADE/GG). All required information how to create a datatype were obtained from sample source code by David N. Junod found in the 3.1 NDK.

The asynchronous file I/O functions used are taken from an example file of the original picture.datatype V43. It was written by Matthias Scheler who allowed me to use his 'ffr.c'. I made some changes to adapt it to my needs, to add save capabilities and to get it compile with GNU CC cleanly.

I have to thank Francis Labrie for his interest in this datatype and Deok-Min Yun who tracked down errors in the encoder part, tested several intermediate versions and added the initial HAM conversion functionality.

The class init code is loosely based on the class initialization from the AIFF datatype 1.16 by Olaf 'Olsen' Barthel. The HAM6 and HAM8 conversion functions are derived from assembler functions written by Olaf.

Futhermore, a thanks must go to Roland Mainz who pushes further datatypes development and supplies useful example codes.

## 1.10 history

- 40.11 (19/09/98) - V42MODE was always active for a valid HAM prefs entry
    - reduced stack usage during load for safety reasons ;(
    - fixed grayscaling of paletted images (RED and BLUE scaling was swapped by accident...)
    - added support for local preferences
  - 40.10 (20/08/98) - used wrong function for B/W encoding in V42 mode and wrong function for 24bit data in V43 mode :( (nethertheless V43 save mode still doesn't work)
    - fixed broken B/W encoder (was totally messed up)
    - added support for true-colour images in V42 mode
-

- (default: grayscaling, optional HAM6/8 encoding)
  - added conversion of 24bit data into HAM6/8
  - EHB image now saved as paletted image
  - decoder partially rewritten ;-)
  - added a preferences system
- 40.9 (16/06/98) - class initialization partially rewritten (which fixed a stupid bug in LibOpen() - previous asm code was ok)
- added encoder (HAM and EHB pictures saved as 24Bit)
  - several additional tweaks
- 40.8 (03/06/97) - class initialization rewritten in C (hi coto ;-)
- a save request for DTWM\_RAW returns a failure now
- 40.7 (11/04/97) - added missing sanity check for DTA\_SourceType
- 40.6 (19/07/96) - buffer handling for asynchronous I/O improved to gain better performance on 68040/68060 based systems (size adjustment and quad-longword alignment)
- 8bit colour value now spreaded over full 32bit
  - version for 68020+ equipped amigas included
- 40.5 (21/04/96) - added support for compressed 4 and 8 bit images
- added support for the new picture.datatype V43
  - added asynchronous file I/O
  - other optimizations
- 40.4 (11/08/95) [ previous version was released a little bit to fast ]
- forgot to correct the normal address in the guide :(
  - source code cleanup (reduced the executable size :-)
- 40.3 (10/08/95) - fixed a serious bug with 8-bit images (did not use a separate pixel buffer for WritePixelLine8(), but this is *\*absolutely\** required due to a size restriction...)
- replaced AllocVec() with Exec pool-functions of V39+
  - reduced stack usage
- 40.2 (24/06/95) - library bases are now taken directly from the classbase (no additional global library bases required anymore)
- fixed a (harmless) bug in the bmpheader decode function
  - freed a buffer in the image decoder at a wrong place...
- 40.1 (22/05/95) - initial release

## 1.11 author

email: gnikl@informatik.uni-rostock.de

or

irc: munk on #amigager

or

snail: Gunther Nikl

---



Ziegenderfer Chaussee 96  
Parchim  
19370  
GERMANY