BMPdt

Gunther Nikl

COLLABORATORS			
	TITLE : BMPdt		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Gunther Nikl	April 25, 2025	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

BMPdt

1.1 BMP picture datatype

```
BMP Picture DataType for Workbench 3.0 or above
      Written by Gunther Nikl in 1995/96
                FreeWare

          Introduction
          Features
          Installation
          Acknowledgments
          History
          Author
```

1.2 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 has been created to supersede the bmp.datatype v39.4 from the 3.1 NDK or pictdt_42_1.lha. That datatype was limited to 16 and 256 color 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 if it is indeed possible to use GNU-C to program one :^)

1.3 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 the 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

Currently displaying true-colour images requires the picture.datatype V43.
With older versions of the picture.datatype one will get an error message
saying "ERROR_NOT_IMPLEMENTED". This may change in a future version :-)

The ECS/AGA support of the new picture.datatype is currently broken when a
custom screen is used as output device for non-truecolor pictures. However,
displaying those images in a wb window works fine. True-color pictures are
not affected at all.

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

1.4 installation

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

- Classes/DataTypes/bmp.datatype

- Devs/DataTypes/Windows BMP
- Devs/DataTypes/Windows BMP.info

- BMPdt.guide
- BMPdt.guide.info

- Source code

Copy the "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).

1.5 acknowledgments

This "BMP" datatype was written from scratch using GNU-C 2.7.0 for the C-part
and SNMA for the assembler part. All required information how to create a
datatype were obtained from the sample source code by David N. Junod found in
the 3.1 NDK.

The asynchronous file I/O functions used had been adapted from an example file
of the new picture.datatype V43. It was written by Matthias Scheler who allowed
me to use his 'ffr.c' with this datatype. I made some changes to adapt it to my
needs and to get it work with GNU-C.

1.6 history

```
v40.5 - added support for compresssed 4 and 8 bit images
      - added support for the new picture.datatype V43
      - added asynchronous file I/O
      - other optimizations

v40.4 [ 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 :-)

v40.3 - 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...)
      - switched from AllocVec() to exec pool-functions of V39+
      - reduced stack usage

v40.2 - 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...

v40.1 - initial release
```

1.7 author


```
  email: gnikl@informatik.uni-rostock.de

         or

  snail: Gunther Nikl
         Ziegendorfer Chaussee 96
         Parchim
         19370
         GERMANY
```


```
Final note: Use at your own risk!
```

1.8 gcc

```
Altough GCC is not a very amiga specific compiler compared to SAS/C it
can be used to write amiga only programs as this datatype should show.
GCC doesn't offer registerized function calls so another approach has
to be used. Therefore the initialization part of the datatype was written
in assembler, the actual image loading and decoding functions in C. The
C part needed some library bases to be "global" initialized by the asm
part. The class dispatcher calls the C function with the actual classbase
in a specific processor register. I used "A4" but every other non-scratch
register could be used (except A5, A6 or SP :). With GCC one can use this
classbase by declaring a *global* register variable. The used register
will be unavailable during the complete compile time and therefore always
contain the address of the classbase that holds all required library bases.
```

All one has to do is to ensure that library bases are taken from there by redirecting some #defines to these entries. Please consult the soure code to see how its actually done.