

JFIFdt

Gunther Nikl

COLLABORATORS

	TITLE : JFIFdt		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY	Gunther Nikl	July 31, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	JFIFdt	1
1.1	JFIF picture datatype	1
1.2	copyright	1
1.3	disclaimer	2
1.4	introduction	2
1.5	features	2
1.6	installation	2
1.7	preferences	3
1.8	faq	5
1.9	acknowledgments	5
1.10	history	6
1.11	future	7
1.12	author	7

Chapter 1

JFIFdt

1.1 JFIF picture datatype

JFIF Picture Data Type for Workbench 3.0 or above
Written by Gunther Nikl in 1998-2001

GiftWare

Disclaimer
Introduction
Features
Installation
Preferences
Datatype FAQ
Acknowledgments
History
Future
Author

Final note: Use at your own risk!

1.2 copyright

This JFIF package is Copyright © 1998-2001 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.

If you use this datatype you should consider to send the author a small gift.

This software is based in part on the work of the Independent JPEG Group.

libjpeg 6b - March 27, 1998
Copyright (C) 1991-1998, Thomas G. Lane.

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 jfif.datatype can be used with either an old V42 style picture.datatype or with a new V43 one. It adapts itself to the underlying version of the picture.datatype. Since this datatype currently doesn't support secondary backing store, one should have enough physical memory available.

I wrote this datatype because I wasn't satisfied with the ones available ;-)

This datatype was tested on an A4000/060 with a PicassoIV running OS3.5 + P96.

1.5 features

- supports normal and progressive JFIFs with all its flavours
- supports HAM output (HAM6 as well as HAM8)
- 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
- automatic colour quantizing of true-colour data for V42
- can be configured to suit personal preferences
- OS 3.5 support (disables dithering for hi/true colour screens)
- FBlit support (affects V42 mode only)
- uses memory pools
- based on libjpeg 6b

1.6 installation

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

- Classes/DataTypes/680x0/jfif.datatype
- Devs/DataTypes/JFIF
- Devs/DataTypes/JFIF.info

- JFIFdt.guide
- JFIFdt.guide.info
- Source code

Copy an appropriate "jfif.datatype" into the "SYS:Classes/DataTypes" drawer. The file "JFIF" and its info file should be placed in the "DEVS:DataTypes" drawer. In order to use the datatype doubleclick on "JFIF.info" (or reboot the machine). If there was another jfif.datatype installed its descriptor has to be removed from the "DEVS:Datatypes" drawer. A reboot is required in this case to get the new datatype activated.

Note: All versions compiled for >= 68020 do *require* a FPU!

1.7 preferences

The datatype can be configured by placing desired options in an environment variable. The name of this variable is "classes/datatypes/jfif44.prefs". Please use Setenv (or Echo if you prefer) to create *global* or 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,UNSAFE/S,DEPTH/K/N,SCALE/K/N,DITHER/K,DCT/K,
FLIPX/S,FLIPY/S,IGNORE/S,VERBOSE/S
```

```
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. This is significantly faster than colourmapped output.

```
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 silently ignored when working in grayscale mode or if the supplied argument is invalid.

```
UNSAFE
-----
```

When working in V42 mode this option enables a FBlit related optimization provided FBlit is installed.

```
DEPTH
-----
```

This argument allows you to select the number of palette colours. Valid inputs are between 3 and 8 giving between 8 and 256 colours. Anything >8 selects true-colour output (24bit) but only if the datatype works in V43 mode and grayscaling is off. If 24bit output isn't possible the datatype switches to 256 colour palette output.

Note: If HAM mode is selected a given DEPTH option is **ignored**.

SCALE:

This option requests the datatype to scale the image by 1/1, 1/2, 1/4 or 1/8. The scale factor may be 0..3 which specifies the scale as exponent.

DITHER

To improve the image quality for **palette based** output a dither method may be choosen:

- NONE - no dithering (bad quality,fast)
- ORDERED - ordered dithering (better quality,slower)
- FS - floyd-steinerberg dithering (best quality,slow)

The default is to use floyd-steinerberg dithering.

DCT

Selects the decode method to be used. Possible choices are:

- FAST - fast, but less accurate integer algorithm
- SLOW - slow, but accurate integer algorithm
- FLOAT - slow, but accurate floating-point algorithm (*)

The default algorithm choosen is 'FAST' if nothing was specified.

(*) The floating-point algorithm is only available in a datatype version especially compiled to support a math coprocessor!

FLIPX,FLIPY

When specified the switches cause the image to be mirrored in X direction (horizontally) and/or Y direction (vertically). Although this is only a datatype these options may be quite useful.

IGNORE

Requests the datatype to ignore all non-memory related decoding errors.

NOTE: This switch does **only** affect the error handling of libjpeg!

VERBOSE

This switch (currently) affects the behaviour of error messages. By default no error message is displayed in case something goes wrong. When this jfif datatype is in "verbose" mode there might be a requester to show the error message since the requester will be suppressed if the calling process does not want a message displayed. The message should appear on the callers

preferred screen (if there is one) rather than on the default public screen.

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!

Q: Why is the floating-point dct algorithm not supported with non-fpu version of this datatype?

A: Floating-point support on Amigas not equipped with a math coprocessor is done either through IEEE or FFP math libraries. The FFP libraries can be shared between multiple users but the FFP format isn't supported by GCC. Therefore one has to choose the IEEE libraries for floating-point support. Unfortunately these libraries require that every user opens them itself. Since the base addresses of these libraries have to be accessed as FAR variables there is only one variable possible thus the floating-point dct algorithm was omitted for non-fpu versions of this datatype.

The quality of the slow-integer dct algorithm should be sufficient though.

1.9 acknowledgments

This "JFIF" datatype was written from scratch using GNU CC 2.7.2.1 and 2.95.2 (ADE/GG) for 000/020+FPU/040+FPU versions and SAS/C 6.58 for the 060+FPU version. All required information how to create a datatype were obtained from sample source code by David N. Junod found in the 3.1 NDK.

This datatype is based on the work of the Independent JPEG Group (IJG). Without their work the datatype wouldn't exist at all.

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 Frank Wille for pointing out a problem I was unaware of. This datatype used quite some stackspace and the original IPrefs has very little of it thus crashing when loading a picture. Since my IPrefs runs with more stack I didn't noticed the problem...

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 pushed further datatypes development in the past and supplied useful example codes.

Finally, I have to thank Timm S. Mueller (author of eg. MysticView) for being the first one sending a small gift. Congratulation ;-)

1.10 history

- 44.7 (09/09/2001) - added much improved datatype descriptor for better jpeg recognition (Exif, Mavi, whatever ;-)
 - disabled own save handling to make Multiview happy (Luca 'Hexaae' Longone)
 - 44.6 (03/03/2001) - modified library initialization to fix some holes according to hints from Thomas 'Thor' Richter
 - added FBlit support for V42 mode (Luca 'Hexaae' Longone)
 - 44.5 (17/09/2000) - added (probably) missing WaitBlit()s in V42 mode
 - changed method handling in V43 datatype mode (switched from vararg to "normal" method call)
 - disables dithering on hi/truecolour screens if picture.datatype V44 is detected (-> OS 3.5+)
 - refined jpeg decoding options to gain speed
 - added ability to ignore image errors
 - added a scoptions file
 - added color icons ;-)
 - recompiled with gcc 2.95.2 and SAS/C 6.58
 - (18/02/2000) - adaption for V44 includes
 - 44.4 (08/11/1998) - fixed serious bug in mem copy/clear functions (never occurred but would crash very badly :-)
 - several minor changes
 - 44.3 (19/09/1998) - added support for local preferences
 - reduced stack usage (I hate IPrefs)
 - recompiled with EGCS 1.0.3a / 1.1b
 - disabled abort feature
 - 44.2 (18/08/1998) - added conversion of 24bit data into HAM6/8
 - added decoding abort when detecting Ctrl-E
 - changed handling of grayscale pictures
 - several minor improvements
 - 44.1 (17/06/1998) - initial release
-

1.11 future

Things that might be added in a future release:

- support for reading comment markers
- support for secondary backing store
- localization of error messages
- a PPC version
- an encoder
- ...

The future of this program depends on YOU!

1.12 author

IMPORTANT, my current email can be found in: AmiNet:/docs/anno/munk.lha

email: gnikl@informatik.uni-rostock.de

or

www: <http://home.pages.de/~munk/>

or

irc: munk on #amigager

or

snail: Gunther Nikl
Ziegendorfer Chaussee 96
Parchim
19370
GERMANY