

TIFF

COLLABORATORS

	<i>TITLE :</i> TIFF		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		June 15, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	TIFF	1
1.1	TIFF Data Type Guide 1.01 Copyright © 1993-95 Bert Wynants	1
1.2	TIFF Data Type Documentation - Introduction	1
1.3	TIFF Data Type Documentation - Requirements	2
1.4	TIFF Data Type Documentation - Installation	2
1.5	TIFF Data Type Documentation - About the program	2
1.6	How to use it	2
1.7	dither options	3
1.8	Gray Image Conversion Options	3
1.9	Color Image Conversion Options	4
1.10	TIFF Data Type Documentation - About the Author	4
1.11	TIFF Data Type Documentation - Known Bugs	5
1.12	Introducing TIFFView	5
1.13	TIFF Files	5
1.14	MacPaint Files	5
1.15	GIF Files	5
1.16	IFF Files	6
1.17	JPEG Files	6
1.18	TIFF Data Type Documentation - History	6

Chapter 1

TIFF

1.1 TIFF DataType Guide 1.01 Copyright © 1993-95 Bert Wynants

Table of Contents:

Welcome to the TIFF datatype AmigaGuide®. Select any of the following topics:

[Introduction](#)

[System requirements](#)

[Installation](#)

[Usage](#)

[About the program](#)

[About the author](#)

[Known Bugs](#)

[Program History](#)

1.2 TIFF DataType Documentation - Introduction

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 utilizes these datatypes and displays any file for which you have a data types class installed. Already supplied on your Workbench disk is the ILBM datatype for loading regular IFF-ILBM pictures.

Using this TIFF datatype you can enhance the functionality of MultiView and other datatypes-aware applications.

TIFF Files (Tag Image File Format) are files commonly used in the graphic computer world (Mac, Sun, NeXT, IBM)

this TIFF Datatype is compatible with the 6.0 specs of TIFF.

supported TIFF:

-multiple image TIFF

-1,2,3,4,5,6,7,8,24,32 Bit

-Line-art, gray, palette images, 24 bit color RGB, 32 Bit color CMYK.

1.3 TIFF DataType Documentation - Requirements

Requirements

To use datatypes, you need Amiga OS Release 3. Older versions will not be sufficient.

You should have 2 megabytes main memory.

The installation script requires the Commodore Installer utility. It is not included.

Should you have any trouble running the TIFF Datatype on your machine, please **write to me** with the full specifications of your machine, that is KickStart version, model, expansion boards etc...

1.4 TIFF DataType Documentation - Installation

Installation

In order to get the TIFF datatype installed properly, you should use the supplied installation script. If you do not have the Commodore Installer you will have to install the datatype manually. Install it the same way the Installer would do it.

Manual installation is disrecommended.

1.5 TIFF DataType Documentation - About the program

Current version 40.2

This program is freeware, this means that you can copy it freely as long as you don't ask any more money for it than a nominal fee for copying. If you want to distribute this program you should keep this document with it. This program cannot be used for commercial purposes without written permission from the **author**.

The TIFF Datatype has proven to be stable in everyday use. The author is not responsible for any loss of data, damages to software or hardware that may result directly or indirectly from the use of this program. The author reserves the right to make changes to the software or documentation without notice.

The TIFF Datatype was written in C and compiled using SAS/C 6.55.

None of the files of the TIFF Datatype package may be modified. Crunching or achieving is allowed only if none of the TIFF files get modified by it.

1.6 How to use it

If the TIFF datatype is installed and activated, usage is straight forward. Every program that can work with picture datatypes can open TIFF files also.

The Preference Editor

This editor is used to set some options of the TIFF datatype and should be installed in the systems Prefs Folder.

The options:

Dithering

Gray Image

Color Image

1.7 dither options

Dithering

The dithering algorithm used in the datatype is called error diffusion. It is a method to make images look good on devices with less colors than the image really contains. It does this by remembering the error on a pixel and spreading that error over the next pixels.

No Dithering

Dithering is disabled.

Simple

Just a simple and fast one dimensional dithering.

Floyd & Steinberg

Good and fast 2-dimensional dithering.

Jarvis, Judice & Ninke

Slow dithering spreading error over 3 lines.

Stucki

Slow dithering spreading error over 3 lines.

Stevenson & Arce

Slow dithering spreading error over 5 lines.

1.8 Gray Image Conversion Options

Gray Image

Normal gray TIFF files are 4, or 8 bit deep (variations are possible, however not common).

Convert To Line-Art

The image will be converted to a line-art image using the selected **dithering** algorithm.

Limit depth to 2 bit

If the TIFF image has more then 2 bits it will be converted to a 2 bit image using the selected **dithering** algorithm.

Limit depth to 4 bit

If the TIFF image has more then 4 bits it will be converted to a 4 bit image using the selected **dithering** algorithm.

Full image Depth

The depth of the image is as specified in the file.

1.9 Color Image Conversion Options

Color Image

Normal palette TIFF files are 1,2,4 or 8 bit deep.

Normal RGB color TIFF files are 24 bit deep.

Normal CMYK color TIFF files are 32 bit deep.

(variations are possible, however not common)

Convert To Gray

The image will be converted to gray with a depth as specified in the **Gray Image** options.

Limit depth to 3 bit

If the color TIFF image has more then 3 bits it will be converted to a 3 bit image using the selected **dithering** algorithm. This is 1 bit red, 1 bit green, 1 bit blue.

Limit depth to 6 bit

If the color TIFF image has more then 6 bits it will be converted to a 6 bit image using the selected **dithering** algorithm. This is 2 bit red, 2 bit green, 2 bit blue.

Limit depth to 8 bit

If the color TIFF image has more then 8 bits it will be converted to a 8 bit image using the selected **dithering** algorithm. This is 3 bit red, 3 bit green, 2 bit blue.

1.10 TIFF Data Type Documentation - About the Author

If you have suggestions or remarks about this program, or if you find any bugs, please let me know.

When sending in bug reports, please state exactly under what circumstances the bug occurred, what equipment was used and what happened. If possible also try to give me enough information to reproduce the bug. It is very difficult to find bugs when you don't know exactly what happened.

Write to the following address:

email : bert.wynants@innet.be

Fido : 2:292/603.76 (Bert Wynants)

SnailMail: Wynants Bert

Hogeheide 6

B-2260 Heultje-Westerlo

Belgium - Europe

1.11 TIFF DataType Documentation - Known Bugs

Known Bugs

This datatype has problems reading 3 bit-TIFF files from BME, a program of Softlogik (belongs to the Pagestream 3h-distribution).

This is no bug in the datatype, but a problem in the BME writer.

1) They claim a highly compatible TIFF file, but a 3 bit Palette TIFF file is not standard according to the TIFF specification.

2) They use an illegal packing in the data.

3) This datatype can work with 3 bit TIFF files saved by **TIFFView** ,

And these 3 bit TIFF files also read correctly in Adobe PhotoShop.

1.12 Introducing TIFFView

This is a utility to read, view and print **TIFF** , **MacPaint** , **GIF** & **IFF** , **JPEG** or any image you have a DataType for, and to write them back as **TIFF** , **JPEG** or **IFF** files.

This is also one of **my** programs.

1.13 TIFF Files

TIFF Files (Tag Image File Format) are files commonly used in the graphic computer world (Mac, Sun, NeXT, IBM)

TIFFView is compatible with the 6.0 specs of TIFF.

supported TIFF:

-multiple image TIFF

-1,2,3,4,5,6,7,8,24,32 Bit

-Line-art, gray, palette images, 24 bit color RGB, 32 Bit color CMYK.

1.14 MacPaint Files

MacPaint Files are that old files used on Macintosh. I included this format because some clip-art libraries still use this format...

1.15 GIF Files

GIF Files (Graphics Interchange Format(c)) are files that are commonly used in the BBS and IBM World it is however limited to 256 colors.

supported GIF: GIF87a & GIF89a

-multiple image GIF files

-1,2,3,4,5,6,7,8 Bit interlaced & non interlaced files

-Line-art, gray, palette images.

1.16 IFF Files

IFF Files (Interchange File Format) are files that are commonly used in the Amiga World.

supported IFF:

-1,2,3,4,5,6,7,8 Bit

-Line-art, gray, palette images.

1.17 JPEG Files

JPEG (pronounced "jay-peg") is a standardized compression method for full-color and gray-scale images. JPEG is designed to handle "real-world" scenes, for example scanned photographs. Cartoons, line drawings, and other non-realistic images are not JPEG's strong suit; on that sort of material you may get poor image quality and/or little compression.

JPEG is lossy, meaning that the output image is not necessarily identical to the input image. Hence you should not use JPEG if you have to have identical output bits. However, on typical real-world images, very good compression levels can be obtained with no visible change, and amazingly high compression is possible if you can tolerate a low-quality image. You can trade off image quality against file size by adjusting the compressor's "quality" setting.

1.18 TIFF DataType Documentation - History

History

Version 40.1

-initial release.

Version 40.2

-fixed AllocBitMap BMF_DISPLAYABLE (thanks Gunther Nikl for reporting)

-Added preference editor, this allows several options to be set

for the datatype.

Version 40.3

-fixed bug in preference editor. it saves the settings OK but displays them

incorrect on reopening (thanks Stanley Tomshinsky and Mark A. Gotz for reporting)

-Added a proper icon for the preference editor (see in prefs drawer for a Magic WB version).

-Small fix in installer script.

-If the requested Public Screen is not available it falls back to the default Public Screen.

-Added german catalog (thanks to Lars Eilebrecht and Dieter Stolte).
