

akTIFF_Documentation

COLLABORATORS

	<i>TITLE :</i> akTIFF_Documentation		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		May 24, 2025	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	akTIFF_Documentation	1
1.1	akTIFF : Documentation	1
1.2	copyright	2
1.3	disclaimer	3
1.4	distribution	3
1.5	payment	4
1.6	Usage and so on	4
1.7	Datatype FAQ	5
1.8	Making use of 680x0 CPUs and PPC accelerators	9
1.9	correspondence	12
1.10	thanks	13
1.11	prefs	14
1.12	history	18

Chapter 1

akTIFF_Documentation

1.1 akTIFF : Documentation

akTIFF.datatype V44.43

- SHAREWARE -

© 1998-99 by Andreas Ralph Kleinert. All rights reserved.

A PerSuaSiVe SoftWorX PRODUCT.

Needs Kickstart V3.x

Release Date : 18.07.1999

Please consider registration - usually less than 1% of the
users of a program do register. That's not much.

<Commercial> BTW: What is SViewII ? </Commercial>

Copyright
Disclaimer
Distribution
Payment
Usage and Notes
Datatype FAQ
68020-68060, PPC
Prefs
Correspondence
Hall of Fame
Version-History

_ //
Only \X/ Amiga makes it possible!

Please visit:

WWW Support Site
<http://www.ar-kleinert.de> (AWeb-II)

The CHAOS theory:

"Like finding that bloody butterfly whose flapping wings cause all these storms we've been having lately and getting it to stop." (see "Witches Abroad" by Terry Pratchett)

Ahm...well:

...and thanks for all the fish.

1.2 copyright

The akTIFF.datatype in this version and its documentation files are (C)opyright 1998-99 by Andreas R. Kleinert. All rights reserved.

The right of using this program is granted to you by paying the SHAREWARE-fee of 15 DEM (10 U\$) or equivalent (e.g. in Euro) to the author.

This software is based in part on the TIFF reference library (libtiff), which allows being used e.g. for freely distributable and commercial programs.

libtiff 3.4beta037:

Copyright (c) 1988-1997 Sam Leffler
Copyright (c) 1991-1997 Silicon Graphics, Inc.

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that (i) the above copyright notices and this permission notice appear in all copies of the software and related documentation, and (ii) the names of Sam Leffler and Silicon Graphics may not be used in any advertising or publicity relating to the software without the specific, prior written permission of Sam Leffler and Silicon Graphics.

THE SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

IN NO EVENT SHALL SAM LEFFLER OR SILICON GRAPHICS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Note, that reading of LZW compressed TIFF graphics is disabled because of legal reasons. Zip and JPEG compression intentionally is not supported, because they are seldomly used and support would bloaten the binaries

a lot - subject to further considerations, though.

akDT_Installer by Robert C. Reiswig ©1996-1998.

If you wish to use any part of this installer you must ask. May not be integrated/placed into any other package! Changes, suggestions or problems: akDatatype@vgr.com

WarpUP Elfloader (ElfLoadWOS) code originally by Peter Annuss <paladin@cs.tu-berlin.de> which it is needed for loading/executing EGCS 2.91.57 WOS PPC binaries under AmigaOS (see <http://cs.tu-berlin.de/~paladin/> for further information). Has been completely rewritten and quite somewhat enhanced and bugfixed in the meantime, though.

Prefs GUI design improved by Georg Rottlaender <Georg.Rottlaender@bonn.netsurf.de> under use of a 'NewIcon' graphics by Philip Vedovatti <vedovatt@u.washington.edu> - included with kind permission by the 'Team NewIcons'

The patch files were created using the scompare SAS Binary File Compare Program V6.50 which is copyright © 1992-1993 SAS Institute, Inc. The spatch SAS Binary File Patcher V6.50 is copyright © 1992 SAS Institute, Inc.

Some of the mentioned names or products within this or other documents may be copyrighted by companies or trademarks of companies or persons.

Should any of the listed terms and clauses within this document not be valid in conjunction with the law of certain countries this does not affect the validity of the other clauses.

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.

The author reserves the right to discontinue development of the program.

1.4 distribution

The akTIFF.datatype in this version is freely distributable (SHAREWARE). You may copy it, if the copyright notice is left intact and all of its parts are included in the distribution.

This program may only be included in commercial packages or commercial program collections with my written permission - ask for it.

This program may be put on public domain disks or included in public

domain disk libraries - when being distributed that way, it is allowed to take a nominal fee including the costs for copying, without considering that as "commercial" in the above mentioned sense.

This program may also be distributed via electronic mail and may be put into mailboxes as long as the redistribution conditions are respected in all points.

By using or distributing this program you automatically agree to all of the above conditions and terms.

1.5 payment

You may send cash money in an envelope, euro-cheques, or just transfer the 15 DEM (10 US\$) shareware fee to the following account (mention your name): Deutsche Bank Siegen, BLZ 46070090 Kto. 0298174

No foreign cheques, please (euro-cheques or DM-cheques are ok).

1.6 Usage and so on

Installation and Usage

Just install the datatype files to their appropriate directories, and copy the akTIFFPrefs command to SYS:Prefs/Datatypes (optionally).

While the datatype itself can be placed elsewhere within a valid search path, the .ppc module HAS TO be placed to SYS:Classes/Datatypes/ - not a problem, if you use the installer script, otherwise please remember...

Program information

akTIFF.datatype is a TIFF datatype, which is based on the latest TIFF reference sources (libtiff 3.4beta037).

So it does support 8 Bit color mapped files (colorspace expanded to 8 bit per component always) and True color files (24 Bit, alpha channel ignored, cut down to 24 Bit 8:8:8 if necessary).

Note, that reading of LZW compressed TIFF graphics is disabled because of legal reasons. Zip and JPEG compression intentionally is not supported, because they are seldomly used and support would bloaten the binaries a lot - subject to further considerations, though.

With V39-V42 picture.datatype it either produces (upto) 256 color palette-based or HAM6/8 output (256 colors exported unmodified, 24 Bit data either dithered or converted to HAM6/HAM8) with picture.datatype V43 as well 24 Bit may be exported unmodified.

There are picture.datatype V43 versions available for both, CyberGraphX and Picasso96, while the one for Picasso96 does work

with ECS/AGA, too - simply use the appropriate one.

You must use the included preferences program for best configuration - of course you can also use one of the alternative prefs programs from Aminet, which should deliver the same functionality (but please remember not to send any corresponding bug reports to my address).

akTIFF.datatype is SHAREWARE, the future depends on YOU.

1.7 Datatype FAQ

"Object is not of required type"

Note, that reading of LZW compressed TIFF graphics is disabled because of legal reasons. Zip and JPEG compression intentionally is not supported, because they are seldomly used and support would bloaten the binaries a lot - subject to further considerations, though.

"Not enough memory"

The main reason why this datatype has been created was to supply a PPC-optimized TIFF datatype. 68k support has been added for completeness (and as fallback option), however it uses the same base construction as the PPC version and thus uses much more memory than necessary - however this may speed up loading somewhat, even in the 68k version (compared to other TIFF datatypes).

This datatype isn't suitable for 2 or 4 MB machines - you should have some fastram in spare - otherwise, use some of the competing TIFF datatypes.

CTRL-E support ?

No, not this way, mate !

Keyfile system

There's a keyfile system used for this datatype - note, that the keyfile actually does not enable any "extra functionality" except making the 68k and PPC module fully functional and just replacing that "Registered ?" text in the progressbar.

I won't send any keyfiles via snail mail. If you want to receive the key, please mention your email address (clearly written) with your registration !

NOTE: keyfile can be placed to either S: or where KEYPATH (env-variable) does point to.

PPC module (WOS)

This one is experimental and follows nearly exactly the same rules as the one for PPC - it just is named "akTIFF.wos" (290K) and uses powerpc.library V14+ instead.

The external program "C:LoadElfWos" will be used for running the PPC ELF module (with speed penalties!) unless LOADELF_WOS=OFF has been set in the preferences file.

Remarks for LOADELF_WOS=ON:

Maybe making "C:LoadElfWos" resident (set the "p" bit and say "Resident C:LoadElfWos" in s:user-startup) may give a little speed-up. However, you need a version of C:LoadElfWos that actually can be made resident. Maybe you'd simply like to try that out...

Remarks for LOADELF_WOS=OFF:

In case LOADELF_WOS=OFF has been set, stability problems (*) with some programs may occur (e.g. with dopus_pattern or WBPatten). Program specific settings may make sense here (e.g. explicitly use LOADELF_WOS=ON for these programs, but set it to LOADELF_WOS=OFF for others).

Using the CACHE_WOS option will avoid re-loading of the ELF module from disk every time when it is invoked, but instead keeps it in memory all the time (needs twice as much memory, even during the decoding process, but should be noticeably faster). CACHE_WOS setting may be changed during run-time.

CACHE_WOS=ON is recommended, if you want the highest possible speed and don't care so much about memory usage - however be careful, if memory gets too low, the TIFF loader may fail, which again will mean an even bigger disadvantage. This is especially important, because the WOS version - for a short moment - needs twice as much memory when transferring the image from the PPC to the 68k side (this has technical reasons). (*) reason: unknown

Last words:

The datatype's ELF module for ppc.library basically already do work with the latest beta version of Frank Wille's ppc.library emulation for WOS (V0.6b or higher) - I'd recommend to simply try out, which version does run faster: the native WOS version or the emulated PPC version. Since the PPC version does not need "C:LoadElfWos", this is an open question.

The latest ppc.library emulation for WOS can be found on Frank Wille's homepage under <http://home.owl.de/~frank/>

PPC module (ELF)

Yes, this datatype is prepared for a great speed up with phase5's powerUP (TM) boards.

For this, the ELF TIFF decoder module has to be placed at location SYS:Classes/Datatypes/akTIFF.ppc - the installer script will manage this for you on demand.

Make sure that you've the 68040/060 versions of the datatype installed, since the 68000/030 versions don't contain the necessary

extra code (there are no powerUP boards with 68000/030s CPU available or planned as far as I know). Also, don't install the ELF module and/or ppc.library if you don't have a PPC board plugged in.

Raw loading speed up should be very impressive with this PPC module, although it of course can't increase rendering or dithering (remapping) speed of other system modules or the calling program.

HAM conversion or ordered dithering (for 24 bit images, i.e. if not in V43 mode) are NOT yet PPC optimized - get a graphics card !

Please note, that the datatype (68k AND PPC) only will become fully functional for registered users of this datatype, who have a keyfile installed.

If you don't have a keyfile installed, you have two choices:

1. remove it again
2. make use of the 68k or PPC module but get only every 3rd line of the image (the whole image will be loaded and decoded, but only every 3rd line will be passed to the caller)

Speed: to test the speed of the decoder, you should go online with AWeb and load a WWW page with several large TIFF graphics. Then go offline again, and load the same page from the cache: this will show you the raw decoding speed, without any influence of download time or other tasks.

Best is, to do the speed tests in V40 mode when using the demo version, since in V43 mode, the demo restrictions themselves (= not exporting every line of the image) will have some (undetermined) influence on speed - those lines explicitly have to be *cleared*, which needs some time on a 24 bit image. Sorry - this was introduced after V44.2 with a bugfix.

NOTE: decoding will need about twice as much memory as with a conventional datatype plus for the PPC version approximately another 245K for the loaded ELF module, 16K for stack and 16K for I/O buffers (you know, RISC is 'reduced instruction set' and not 'reduced memory usage' - but now you are able to actually make use of all that expensive RAM ;-)
Also, the progressbar is not available for PPC decoding (does not make much sense when e.g. WWW browsing, anyway).

Small PPC FAQ

Q: Why is a 060/PPC combo faster than the 040/PPC combo ?
A: Perhaps because the 060 can process the I/O requests (aka OS calls) faster than the 040. Small differences may also be caused by using different hard drives - to minimize this, one could put the files into RAM: for example, but this wouldn't deliver real-life results. The following question is related, too.

Q: Can't PPC loaders be faster than this datatype one ?
A: Yes, they actually *can* be faster than the measured results

may indicate. Problem is, that datatypes have to deal with bitmaps, which slows everything down. For example, in 24 bit mode DTM_WRITEPIXELARRAY still has to be performed by the 68k, and in 8 bit mode, the same does apply to WritePixelLine8() - the latter one may include a c2p conversion on systems without a graphics card. To avoid the latter, one for example could try the PPC native loaders for SuperView-Library instead.

More datatypes ?

On Aminet:util/dtype/ you can also find the akJFIF and akPNG datatypes.

No V43 with AGA ?

There's a V43 picture.datatype coming with the Picasso96 RTG package (on Aminet), which works with plain AGA, too.

Crashes ?

The first reason for a crash often is stack size. Not enough stack size.

IPrefs/WBPatterns has this problem, and others as well. Checking this and/or using FastIPrefs (the replacement) is recommended.

For other programs, you may have to increase their stacksize in the program icon or for the CLI/Shell they are called from (e.g. with PPaint).

Using (Fast)IPrefs in PPC mode may not be a good idea at all, but for some people, the following did help in s:startup-sequence:

```
Wait 8 secs
C:FastIPrefs W M L A G
```

For the others, the trick from the Picasso96 FAQ should do the job: put the tool "CPUBlit" (an old patch available on Aminet) to your s:startup-sequence *before* the monitors are started. You must call it as follows:

```
CPUBlit -a -b
```

You may also wish to check out tools like FBlit, FastBlit, CpuBlit98 and related ones from Aminet:util/boot - some may work perfectly on your machines, others perhaps won't at all. But experimenting may be worth it.

No write support ?

Sorry, there won't be write support (DTM_WRITE method), since I think, that datatypes are mainly a system for data exchange and not to do the job of existing conversion utilities.

To explain it even further:

The datatype mechanism certainly is a system to HIDE implementation and data format details. If one does offer too much choices for destination file formats, this would - in my opinion - completely be against this

concept. The ideal way of keeping the datatypes' concept cleanly OOP would be to internally handle everything in an amiga-unique IFF format - which BTW is quite essential for clipboard data exchange as well. Unfortunately IFF-ILBM isn't very suitable for color depths greater than 8 bit. Maybe IFF-RGFX could be a good choice, here.

Odd screenmode selection

graphics.library's BestModeID function isn't so well designed. Try Patching to a better one, e.g. with Aminet:util/sys/ModeP.lha

Progressbar and programs (esp. Browsers)

Please note, that the (optional) progress bar will either open on a windows's screen as specified via pr_WindowPtr, or on the default Public Screen, thus if your favoured Web Browser does not set pr_WindowPtr or does not declare its screen as default pub screen, that's not my fault. PDTA_Screen will be checked first, as well - but usually this won't work at all.

Ramlib Crashes

If you get "ramlib" gurus with this or any other program, then try installing Aminet:util/sys/StackAid.lha

Unknown datatypes (V43)

If your datatypes stop working (unknown file format), please don't blame me, but at first check, whether you've still installed an already expired beta version of picture.datatype V43...

And make sure, that you don't use picdtpatch (v39.2) from the Hypertext.datatype archive by Stefan Ruppert.

Note, that certain TIFF compression types (e.g. LZW) are not supported. This may look as if the TIFF image were not recognized.

1.8 Making use of 680x0 CPUs and PPC accelerators

Basically, this program does run with a plain 68000 CPU.

However, if you do own an 68020/030+68881/882 FPU or 68040/060+FPU, or maybe a dual processor board with PPC, you may wish to make use of the extra horse power.

There are certain configuration options, special libraries and/or patches available, so you perhaps should investigate into that issue a little bit deeper - but carefully.

PPC Support

=====

1. With CyberStorm PPC cards, it may make sense to make use of the "SetFastAvec" and "Set60nsMode" (SetMemMode) tools, which should speed up the system performance somewhat,

i.e. by addressing your RAM with 60ns instead of 70ns access time. Newer versions allow to do these settings fromout the card's bootmenu. If you get random crashes, step back to 70ns.

2. Make sure, that you have a lot of RAM on the accelerator, so that the PPC isn't forced to make accesses to the slow motherboard RAM. If you get random crashes, make sure you followed the installation instructions, and did not configure SIMMs of different vendors for a 64 bit access bank.
3. This program does make use of "ppc.library". So: Make sure, that you a) don't have "powerpc.library" installed or b) have a version of "powerpc.library" installed, which does not conflict with "ppc.library" (V7 is said to work together with ppc.library). Don't install ppc.library without having a PPC board plugged in. Always make use of the newest 68040/68060.library plus ppc.library - as available under ftp.phase5.de or Aminet.

(There's BTW now support for powerpc.library V14 as well, so you can decide. Basically, it even does work to run the PPC-Library version under Frank Wille's ppc.library emulation for WOS [V0.6b or higher].)

4. Read the corresponding FAQ pages for more information on PPC support and configuration - especially note, that a keyfile is required for fully functional PPC support within this datatype.

68020/030+68881/882 FPU and 68040/060+FPU Support

=====

Usually, Amiga OS' mathieeee-Libraries do automatically manage the coprocessor support, but for some reasons, these libraries are not used with this datatype:

- they can't be shared between processes
- they are not actually optimized for 68040/060+FPU as with OS 3.1

Unfortunately, the used FFP libraries don't support an FPU at all.

But there are certain patches available on Aminet, to speed up FPU support in general, add FPU support for the FFP libraries or in general allow more efficient use of the 040/060 CPUs, e.g. by avoiding unnecessary emulation of missing instructions through 68040/68060.library.

Make sure, that those patches don't conflict with certain versions of the 680x0 libraries or even are part of these already. If you've carefully read the docs you may wish to check out the following solutions:

1. Fix bugs within the math libraries

This one has nothing to do with the FFP libraries, but

since there's also a bug in mathieeesingbas.library (which resides in ROM), you should install a patch for that:

- a) best solution is a newer SetPatch Version V43.x
(available from ftp.amiga.de somewhere in "/pub/")
- b) if SetPatch V43 does not work with your OS version,
you should try for example "SetMathPatch"
(coming e.g. with GhostScript - see Aminet:gfx/show)

Those patches may conflict with some math library replacements - it seems to be logically, that a completely rewritten replacement library of course does not need to be patched any further. At least not for the same bugs...

2. Patching the math#? libraries for better (or introducing) FPU support:

- a) - FMath V40.6 Aminet:util/libs/FMath406.LHA
 - FFPPatch Aminet:util/boot/ffppatch.lha
- b) - HSMathLibs Aminet:util/libs/HSMathLibs_040.lha
 Aminet:util/libs/HSMathLibs_060.lha
- c) various other patches from the "util" area of Aminet

With the 68040/68060.libraries of p5, according to their docs, further patches of the math libraries are not recommended - however may work nevertheless.

3. General 040/060 speedup

For automatic speedup on 68020+ systems, this datatype makes use of utility.library.

This one has nothing to do with the FPU, but if you do own a 060 and OS 3.0 you should perhaps consider to install "Mult64Patch", which claims to implement the 64 bit integer functions UMult64/SMult64 of utility.library V39+ (which have to be software emulated on the 060) two times faster than the patches done by 68060.library and four times faster than the trap emulation. A speed test program is included.

That program can be found under Aminet:util/boot/Mult64Patch.lha - however, it may already be obsolete for newer versions of your 68060.library. Do the speed check, then decide.

4. Better performance on 680x0 and PPC

Here, the following tools work quite fine on a 040/PPC board (taken in this order from s:startup-sequence):

```
C:FastExec >NIL: <NIL: NOEXEC FASTSSP FASTVBR FASTEXP FASTMEM FASTINT ↔
REBOOT
C:SetPatch QUIET
C:QuickRom >NIL: <NIL:
```

Run >NIL: <NIL: C:CpuBlit

FastExec V2.9	(Aminet)	-> various speedups
SetPatch V43.6b	(www.amiga.de)	-> OS patches
QuickRom V36.08	(Aminet)	-> ROM to RAM
CpuBlit98	(Aminet)	-> let the CPU do blitting

This all runs fine in 60ns mode, together with SetFastAvec, PPCInstall and CyberGraphX V3.

1.9 correspondence

** General PerSuaSiVe SoftWorX WWW Support Site is <http://wdo.de/ark/>
 ** - actually redirected to <http://www.ar-kleinert.de>

You may reach me the following way.	
Send bug-reports, money or whatever to:	

* SuperView Development & Registration *	
* DRAFU Development & Registration *	
* Image Engineer Registration Site Europe *	
PerSuaSiVe SoftWorX	
Until 30 Sep 99	As With 1 Oct 99
Andreas R. Kleinert	Andreas R. Kleinert
Sandstrasse 1	Am Kornberg 48
D-57072 Siegen	D-57076 Siegen
Germany, Europe	Germany, Europe
+49-271-22869	(to be announced)
(also FAX + AM)	
Weekdays after 18.00h.	
When calling via phone you may leave a message, if I'm not available - but don't expect me calling back to USA, Australia, ... since german phone rates are HIGHLY expensive.	

EMail:

Please send binaries via ARK@News.wwbnet.de, and keep them smaller than 16 KB - otherwise ask before.
 Please think twice before sending them - my postbox is not unlimited in size.

* Do not send binaries via Fido or Fido-Gates ! *

- Fido Andreas Kleinert 2:2457/350.18

- Usenet
 - >>> info@ar-kleinert.de
 - Andreas_Kleinert@gmx.de
 - ARK@News.wwbnet.de
- If nothing else works, try one of these public Fido-Usenet gateways:
 - In Germany:
 - Andreas_Kleinert@p18.f350.n2457.z2.fido.sub.org
 - From USA or elsewhere:
 - Andreas_Kleinert@p18.f350.n2457.z2.fidonet.org

1.10 thanks

Thanks go to (in order of appearance ;-)

=====

- | | | |
|-----------------------|---------------------|----------------------|
| - Robert C. Reiswig | - Georg Rottländer | - Sjord de Vries |
| - Philippe Devilard | - Rune Jensen | - Jürgen Urbanek |
| - Bradley Rogers | - Hal Samuelson | - Antonio Brianese |
| - Sebastian Becker | - Rich Robinson | - Adam Corrano |
| - Beth Hedrick | - Casper Thygesen | - Kai Foelster |
| - Peter Denomy | - Thomas Karlsen | - Luca Baldelli |
| - Leonardo Petrucelli | - Thomas Körner | - Dominique Deangili |
| - Colin Keefe | - Roger Curtis | - Sam Gillies |
| - Paul Kieffer | - Yves Liebercier | - Alan Guillevic |
| - Thomas Lorenz | - Chris Barrow | - Ed Eden |
| - Keith Schyler | - Janko Köhler | - Andrew Mills |
| - Howard Toliver | - Jon Mines | - Magnus Bouvin |
| - Dan Muldin | - Mahieux Pascal | - James Luscombe |
| - Martin Ruston | - William Eaves | - Cameron Snyder |
| - Johnny Nielsen | - Kapryan Kennedy | - Peter Annuss |
| - Larry Urquhart | - Philip Yearbury | - Neil Bowes |
| - Steve Hodson | - Johan Rönblom | - Harald Schulz |
| - Christian Schröpfer | - Michael Fedrowitz | - Denis Zwornarz |
| - Gert Hubers | - Jürgen Seubert | - Frank Müller |
| - Peter Kaltstein | - Peter Theuring | - Kirk Strauser |
| - Telemar Rosenberger | - Phillip Degnan | - Chris Dallimore |
| - Matthew Sawyer | - Simo Koivukoski | - Jeffrey Grzanich |
| - John Hart | - Ian Tyrell | - Pekka Sippola |
| - Frank Böhne | - Petr Voralek | - Antoine Bordier |
| - Patrice Dumont | - Manfred Kern | - Francis Klein |
| - Dominique Harelle | - Arnljot Arntsen | - Havard Lunde |
| - Jürgen Ofner | - Geoff Tovey | - Herve Sonnevile |
| - Sascha Ploss | - Michael Domoney | - Carl Read |
| - James Harrison | - Mark Shaw | - Frank Wille |
| - Adam Suwala | - Winfried Krueger | - Harald Wünsche |
| - Simon J Glover | - Don Cox | - Henrik Jensen |
| - Matteo Consolati | - Jürgen Wilschke | - Stephen Webber |
| - Svein Inge Wik | - Philippe Reux | - Paul Venton |
| - Bjarke Vangsgaard | - Stefan Fischer | - Roberto Muller |
| - Michael Thompson | - Alfred Kendall | - John Orwin |

Thanks also must go to:

- ...all buyers of the SView Productivity Suite from Schatztruhe
- ...the Cloanto team, namely Michael C. Battilana
- ...the people from phase5, namely Ralph Schmidt and Claus Herrmann
- ...the picture datatype V43 programmers, namely Frank Mariak and Olaf Barthel
- ...the other programmers of datatypes, for information exchange and useful comments
- ...dozens of people I forgot to mention here !

1.11 prefs

akTIFFPrefs

akTIFFPrefs is the Preferences Program for akTIFF.datatype.

GUI has been designed with StormWizard 2.0, so this program needs "wizard.library" V37+ (you can find a copy on Aminet under "biz/haage/WizardLibrary.lha" or even newer versions under ftp.haage-partner.com).

Icon by Bert Bosma <lmb@wxs.nl> (based on NewIcons).

An alternative MUI prefs program replacement by Alvaro Thompson (originally) and Achim Stegemann (later) is now available as util/dtype/akMUIPrefs.lha - there also are various other replacements.

The global settings will be written to ENV: (and maybe also ENVARC:) into a preferences file called "Datatypes/akTIFF.prefs".

Task (process) specific settings also can be done - either using the preferences program (which allows to select the corresponding process from a list as long as it actually is running at the same time) or by hand, following the scheme below:

OPTIONAL

----- task specific settings files -----

Settings specific to different caller programs may be created by copying the global settings from "Datatypes/akTIFF.prefs" to an optional task-related prefs file called

"Datatypes/akTIFF.prefs_Tasks/TaSkNaMe"

where "TaSkNaMe" means the name of the program as e.g. shown by a system monitor (for obvious reasons, this does work best with workbench programs, which don't require name patterns as some CLI programs might do, like for example "CLI(3):Work:Browsers/XWebber"). So, with AWeb for example, you would just edit your global settings file and then do the following:

```
MakeDir ENV:Datatypes/akTIFF.prefs_Tasks
Copy ENV:Datatypes/akTIFF.prefs ENV:Datatypes/akTIFF.prefs_Tasks/AWebIP"
```

[... and the same for ENVARC: ...]

After that, AWeb will ignore the global settings and fetch its own from the given file.

You can do the following settings:

- 1) V43_MODE=(NO_DITHERING|V40_DITHERING)
- 2) V40_24BIT_MODE=(DITHER_ORDERED|HAM_OUTPUT)
- 3) V40_DEPTH=(3..8)
- 4) HAM_MODE=(HAM6|HAM8)
- 5) INTERLEAVED_BM8
- 6) DISPLAYABLE_BM8
- 7) PROGRESSBAR=(ON|OFF)
- 8) SPEEDUP
- 9) CUSTOM_MODES
- 10) PPC=(ON|OFF)
- 11) AUTO=(ON|OFF)
- 12) PPCLIB_EMU=(IGNORE|USE)
- 13) CACHEWOS=(ON|OFF)
- 14) LOADELF_WOS=(ON|OFF)
- 15) NOASPECT
- 16) DEBUG
- 17) LZW_ERROR

That's mostly self-explaining, but as an example, here are the default settings and a short explanation:

```
V43_MODE=NO_DITHERING
V40_24BIT_MODE=DITHER_ORDERED
V40_DEPTH=8
HAM_MODE=HAM6
INTERLEAVED_BM8=ON
DISPLAYABLE_BM8=OFF
PROGRESSBAR=ON
AUTO=ON
PPCLIB_EMU=IGNORE
CACHE_WOS=ON
LOADELF_WOS=ON
```

General Explanation of Options

=====

1) V43_MODE

NO_DITHERING: does output 24 Bit data when running pic-dt V43

V40_DITHERING: switches to V40 mode settings when running pic-dt V43

2) V40_24BIT_MODE (when running picture datatype V40 or V43 in V40 mode)

DITHER_ORDERED: does ordered dithering of 24 Bit data

HAM_OUTPUT: does convert 24 Bit data to HAM6/8

3) V40_DEPTH

 When dithering to a palette (so: when in V40 mode and ordered dithering being selected) the number of palette colors, which is 256 by default, may be reduced here (e.g. on ECS systems).
 Valid depth values are 3..8 (which results in 16..256 colors, easily calculated by 2^{depth}).

4) HAM_MODE

HAM6: generates HAM6 output for 24 Bit graphics, when running V39-42
 HAM8: generates HAM8 output for 24 Bit graphics, when running V39-42

Note, that HAM8 is native to AGA machines and thus may cause difficulties with graphic boards and won't work with OCS/ECS Amigas. With HAM6 and graphic boards also problems may occur.

5) INTERLEAVED_BM8

ON: will output interleaved bitmaps upto 256 colors
 OFF: will output normal bitmaps (BMF_CLEAR and maybe BMF_DISPLAYABLE only) - you may switch interleaved mode off for specific programs, which cannot handle it, or when AllocBitmap() has been patched for chunky modes by a graphics card software or e.g. EGSPPlus

6) DISPLAYABLE_BM8

ON: will output displayable bitmaps upto 256 colors
 OFF: will output normal bitmaps (BMF_CLEAR and maybe BMF_INTERLEAVED) - you may turn displayable mode on for specific programs, which want to use datatype generated bitmaps directly as screen bitmap. If they are enabled to do this, this may save some memory (for another bitmap). This is recommended for systems without graphics card and only few chip memory.

7) PROGRESSBAR

ON: pop up percentage display
 OFF: do not pop up percentage display

8) SPEEDUP (hidden option)

Activates some bitmap related optimizations, including a special hack for making image loading with AWeb somewhat faster.

9) CUSTOM_MODES (hidden option)

When the keyword CUSTOM_MODES is set, only viewmodes out of the standard set will be generated:

- LowRes	(320x200/256)
- HighRes	(640x200/256)
- SuperHighRes	(1280x200/256)
- LowRes Lace	(320x400/512)
- HighRes Lace	(640x400/512)
- SuperHighRes Lace	(1280x400/512)

When CUSTOM_MODES=0x##### (e.g. CUSTOM_MODES=0x00000000)

is set, the specified hexadecimal viewmode ID will be used always - alternatively, you can specify the viewmode name as plain text, for example "CUSTOM_MODES=PAL:HighRes". Note, that spelling is very critical here.

For HAM output, this is only true, if the mode ID actually is capable of HAM (this usually is indicated by OR'ing it with HAM_KEY), otherwise a different ID will be computed.

10) PPC (hidden option)

ON: If .ppc or .wos modules are installed, they'll be utilized.
OFF: When the option PPC=OFF is set, the PPC encoder module won't be used, even with a PPC available. Instead the datatype will fall back to 68k mode. Useful e.g. for speed comparisons.

This is a RUNTIME switch. AUTO and PPCLIB_EMU will be processed always.

11) AUTO

ON: Try to find out, which PPC kernel is installed.
OFF: Simply assume, that it's ppc.library

With AUTO=OFF it's not even tried to open powerpc.library.
May cause trouble, if V14+ is installed and gets active sometime (unless we have have a PPCLib emulation running).

12) PPCLIB_EMU

IGNORE: With AUTO=ON and WOS installed, make use of the WOS versions
USE: With AUTO=ON and WOS installed, use the PPCLib emulation

Of course, this only is true for WarpOS' powerpc.library V14+

13) CACHE_WOS

This option is explained in the FAQ.

14) LOADELF_WOS

ON: This will make use of "C:LoadElfWOS" instead of the internal ELF loader code, to avoid some certain problems e.g. with the DOpus viewer or the DOpus/WB background pattern tools. Do not specify CACHE_WOS at the same time (it would be a waste of memory).

OFF: The internal ELF loader code will be used, CACHE_WOS may make sense. If you encounter problems with this option, try increasing the stack of the calling application first (e.g. increase MultiView's stack to 32768 in the icon).

15) NOASPECT (hidden option)

If x/y aspect generation produces buggy results, e.g. with PictIcon, this option may be used to always force 1:1 to be returned.

16) DEBUG (hidden option)

 Will enable debugging output, i.e. open info requesters with concrete information about image size and compression.

17) LZW_ERROR (hidden option)

 LZW compression isn't supported, which usually will cause the datatype system to give an error message like "unknown object" or "not enough memory". If you'd explicitly (by requester) like to be informed when akTIFF does encounter a LZW-compressed image, just specify this option.

If DEBUG is specified, LZW_ERROR will be overridden by a global information message for each image (not just a special one for LZW).

1.12 history

Known Bugs: - some people reported problems with the installation scripts in the past. If you encounter any problems or bugs, please report these directly to the script author Robert C. Reiswig <akDatatype@vgr.com>

- please use at least V41.101 of wizard.library.
You should find a copy coming with demo versions of various programs under ftp.haage-partner.com
- viewmode selection may not always be 'perfect'
(this means, some non-TIFFs might be recognized as TIFFs)
- file recognition is a possible weakpoint

Hint:

- if you use this datatype with a WWW browser, then create a separate partition (sized 30-70 MB) for temporary data storage and do assign VMEM: and your browser's cache directory to it. Also, make sure that it has a decent AddBuffers setting (128 or more). When partitioning (danger: data loss), it may make sense to increase the filesystem block size to a higher value, as well (1024). And make sure, you're using the latest FFS file system 43.x from www.amiga.de - note, that you may update the FFS without repartitioning, but you have to be very careful when doing this fromout HDToolBox.
- even better: use a faster file system (at least) for your cache partition, like the commercial PFS2 (formerly AFS, now by Schatztruhe - see <http://www.schatztruhe.de>) or the free SFS (see <http://www.xs4all.nl/~hjohn/SFS/>)

Keyfile problems:

People, who did not receive their keyfile within 2-4 weeks after sending their registration should also contact me. (During sommer, please note, that it not always does make sense to call after 2 weeks - some people

tend to make holiday sometimes...)

History

=====

- V44.43 (18.07.99): - my postal address will change. Updated docs!
 - progress bar's position and title text changed
 (-> Petr Voralek)

- V44.42 (11.07.99): - added new installer script by Rob Reiswig,
 which should fix the recent "040 gets installed
 with 060" and "devs:datatypes not properly
 installed" problems

- prefs program's icon was missing
 (-> Martin Baute)

- V44.40 (26.06.99): - Aminet release

- V44.39 (16.06.99): - added new installer script by Rob Reiswig

- V44.38 (05.06.99): - fixed "Work:" access

- V44.37 (04.06.99): - prefs: - adjusted PPC part of GUI following
 Georg Rottlaenders suggestions/work
 - there's an Enforcer hit left in akTIFFPrefs
 (don't ask me why, it's not in my code -
 maybe try the MUI prefs or a different
 wiz-lib version instead)

- there again have been reports about some
 faked keyfiles generated by a thing called
 ***** (I won't repeat the name for fame, here).

However, I don't care anymore about these (and especially not about any comments in broken english made in some related .readme files).

There'll always be faked keyfiles and generators for these - if people want me to stop development, they have to continue with that, otherwise they should register. Call it a democratic decision 8-)

Related note: to understand, how PPC and PPC-datatypes do work, one has to understand how PPC works and how the datatypes system in general does work (a good config with pic-dt V43 and actually owning a PPC card plus gfx card may help as well). However, it's not suprising that certain people aren't capable to prove that they a) actually understood it and b) can't do it better (in case I should have been wrong).

A software worth using is worth buying - otherwise simply delete it and "feel better".

- the recommended alternative for WarpOS (WOS) users is, to upgrade to WarpOS V4.0 and install the

latest ppc.library emulation by Frank Wille
(V0.6d or higher, V0.6 will not suffice), then
run the PPC datatype in emulation mode (see prefs).

The latest ppc.library emulation for WOS can be found on
Frank Wille's homepage under <http://home.owl.de/~frank/>

- credit card online registration via RegNet now
is possible. Some special Offers have been
set up for you, some of which are derived from the usual
Discount list.
Or go to <http://www.ar-kleinert.de> to the Amiga Software ←
Area
(RegNet page) and order with only one click! Please have a ←
look!
- please note our new WWW domain www.ar-kleinert.de

V44.36 (30.05.99): - the FAXX datatype now officially has been dropped
from my side; if you're interested in future development,
please contact GPSoft (they got all the sources)
- switched back to LHA 1.38
- Aminet release

V44.35 (24.04.99): - now using LHA 2.1 for generating the distribution
archive (hopefully it'll get smaller, then ;)
- added Rob's newest Installer (thanks again :)
- PPC settings now can be done fromout the preferences
tool (no longer hidden); added several new options
to allow fine-tuning for the used PPC kernel, i.e.
it is no longer necessary to delete the .wos module
if the PPCLib emulation for WOS is to be used
- changed the whole docs accordingly (hey, do you
even read these from time to time ?!)

V44.32 (18.04.99): - WOS: - recompiled using new compiler version
and link libraries; let's see, what happens

V44.31 (10.04.99): - misc changes

V44.30 (23.03.99): - WOS: - completely rewrote, enhanced and bugfixed
internal ELF loader (again)
- reduced loader size to ~2K (LoadElfWos
is ~8K in size)
- and did a special 060 version (which actually
differs from the 040 version now ;)
- also better optimized
- no longer a .lib, but a single .o file

- fixed small, longstanding (possible) bug in progressbar

V44.28 (21.03.99): - WOS: - fixed possible cache problem with LOADELF_WOS=OFF;
maybe this was the cause for unregular crashes
- recompiled using new compiler and new link libs

V44.27 (08.03.99): - fixed several typos in the docs
- added PFS2/AFS, SFS recommendations (for caching)

- use of CMQ no longer recommended (Move16 problem)
 - V44.26 (28.02.99):
 - fixed 32 bit color component extraction, hopefully for all kind of images (-> Denis Zwornarz)
 - added "DEBUG" and "LZW_ERROR" preferences options
 - V44.25 (21.02.99):
 - added some tips and hints on performance optimization to the 680x0/PPC section (list of programs and patches that run fine, here)
 - added note about Frank Wille's ppc.library emulation for WOS
 - various changes to the docs
 - fixed reading on Photoshop 4.0/5.0 written uncompressed 32 bit TIFF files; there was no explicite 32 bit support, but 24 bit would have failed as well... - added/fixed (-> Denis Zwornarz)
 - PPC/WOS: our internal seek derivative for tag scanning was buggy; could cause various problems; more image types should be correctly recognized and decoded now
 - fixed small (harmless ?) bug in II/MM conversion macro
 - V44.24 (12.02.99):
 - DANGER: LOADELF_WOS option now is a switch, either specify LOADELF_WOS=ON or LOADELF_WOS=OFF - "on" is default (this is because so few WOS users seem to do a RTFM before or even after installation)
 - changed docs and FAQ accordingly
 - V44.23 (10.02.99):
 - please upgrade at least to wizard.library 41.101
 - V44.22 (29.01.99):
 - added additional \$VER string to make everybody happy (hopefully)
 - V44.21 (10.01.99):
 - WOS: used fixed EGCS/cwos link library (-> Peter Annuss)
 - V44.20 (09.01.99):
 - WOS: the V44.17 fix actually was MISSING !
 - WOS: fixed another small problem
 - WOS: minor changes
 - V44.19 (07.01.99):
 - fixed LoadElfWOS option
 - V44.18 (06.01.99):
 - added the "LOADELF_WOS" option to work around the DOpus bug (which is still there) (-> various)
 - V44.17 (05.01.99):
 - hopefully fixed DOpus start-bug in akJFIF.wos (-> various)
 - V44.16 (03.01.99):
 - added optional WOS (WarpOS) PPC support: it exactly works the same way as with akJFIF and akPNG
 - V44.15 (01.01.99):
 - new-year cleanup
 - forgot to bump version last time (still was 44.12)
-

-
- V44.14 (22.12.98): - fixed documentation at various places
- V44.12 (19.12.98): - added new installer by Robert C. Reiswig (taking care of any possible .wos and LoadElfWos files)
- added new (hidden) DISPLAYABLE_BM8 option, which is recommended for systems without graphic cards and allows to: save some memory and display larger images (with certain programs, only)
- V44.9 (8.12.98): - prefs program now again compiled using SAS/C instead of StormC: reduced size from 41336 bytes to 22852
- this also fixes possible stack problems with the prefs
- V44.8 (16.11.98): - updated docs
- added PPaint 'stack crash' note
- PPC I/O speedup (similar to akPNG/JFIF)
- V44.7 (5.11.98): - fixed some typos in the docs and related text files
- recompiled prefs program using new StormC link libraries (23-Sep-98). Note, that StormC only is used for the preferences program, not the rest.
- updated docs; revised JPEG/zlib information
- fixed (possible) bug; two 'break's were missing in uncritical section
- added support for uncompressed 4 bit TIFFs (68k/PPC) [added to separate code section, not libtiff-related] (-> Pedersen)
- fixed minor bug (error handling) in sep. code sect. (68k/PPC ←)
- PPC: sep. code sect. for TIFF handling had a major bug
- V44.6 (16.9.98): - first release
-