

akJFIF_Documentation

COLLABORATORS

	<i>TITLE :</i> akJFIF_Documentation		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		July 31, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	akJFIF_Documentation	1
1.1	akJFIF : Documentation	1
1.2	copyright	2
1.3	disclaimer	3
1.4	distribution	3
1.5	payment	3
1.6	Usage and so on	3
1.7	Datatype FAQ	5
1.8	Making use of 680x0 CPUs and PPC accelerators	11
1.9	correspondence	13
1.10	thanks	14
1.11	prefs	17
1.12	history	22
1.13	About PNG - successor of GIF	31
1.14	PNG-Box - WWW tool for PNG writing	31
1.15	JPEG-Box - WWW tool for JPEG writing	32

Chapter 1

akJFIF_Documentation

1.1 akJFIF : Documentation

akJFIF.datatype V44.112

- SHAREWARE -

© 1996-2001 by Andreas Ralph Kleinert. All rights reserved.

A PerSuaSiVe SoftWorX PRODUCT.

Needs Kickstart V3.x

Release Date : 14.08.2001

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

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

Copyright
Disclaimer
Distribution
Payment
Usage and Notes
Free algorithms... PNG and JFIF: a team ...and free speech !
JPEG-Box
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 akJFIF.datatype in this version and its documentation files are
(C)copyright 1996-2001 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 work of the Independent JPEG Group.

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 akJFIF.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 U\$) shareware fee to the following account (mention your name): Deutsche Bank Siegen, BLZ 46070024 Kto. 0298174

SWIFT code for Deutsche Bank Siegen, BLZ 46070024 is DEUTDEDK460.

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

1.6 Usage and so on

GIF is obsolete - you neither should use nor support it any longer. If you are doing WWW design, use PNG and JPEG instead. It's important !

Installation and Usage

Just install the datatype files to their appropriate directories, and copy the akJFIFPrefs 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...

Please make sure, that there is a directory available, where temporary data can be stored. There must be an assignment called "VMEM:" to this directory (just like with SuperViewLibrary and akPNG-dt).

The assignment originally would have been "JPEGTMP:" for the JFIF sources, but the memory manager has been strongly modified and enhanced - on a system with many RAM available you will perhaps never actually encounter usage of that directory.

Do not assign it to "T:" if it is somewhere on a Ram-Disk (that's why T: is not used by default) - just create a safe place for it.

Program information

akJFIF.datatype is a JPEG datatype, which is based on the latest IJG JFIF sources V6b.

So it does support progressive JPEG and all the other things introduced with V6a (note: it reads progressive JPEG, but DataTypes cannot display images progressively - instead only as a single chunk).

With V39-V42 picture.datatype it produces 256 color palette-based (with either none, floyd-steinberg or ordered dithering) or HAM6/8 output, 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.

akJFIF makes use of memory pools where applicable and also automatically utilizes asyncio.library (V39+) when available.

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).

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

1.7 Datatype FAQ

OS 3.5: general note

Well, this datatype DOES work with OS 3.5. With either ppc.library (PPC), powerpc.library (WOS) or ppclib-emu. No matter, what certain people do claim in Usenet. However, certain people did encounter various problems, which have not directly been related to OS 3.5.

For example, with CyberGraphX V4.2 you should make sure to have PLANES2FAST set (with other CGfx versions as well) and sometimes it even may make sense to replace the new picture.datatype V44 with the cgx-based V43 version - because it's simply faster.

OS 3.5 problems

Programs, that let picture.datatype V44 do on-screen dithering, will face the "problem", that 24 bit images even will be dithered when being displayed on 15/16 bit screens. According to the OS 3.5 developer team, this should result in "better image quality".

However, when analyzing this statement, one will discover, that most graphic cards based on PC-chips only allow for 6 bit color lookup tables (LUTs) (that is, 6 bit for each out of red, green and blue - thus only a range of 0..63 instead of 0..255) which in fact isn't much better than the 5:5:5 or 5:6:5 ratio of 15/16 bit high color modes. However, 16 bit high color allows for 65536 distinct colors on screen, while a 6 bit LUT only will allow for 256 out of 262144 colors.

However, these new V44 dithering options can be changed via the datatypes preferences - global default settings then locally will be overridden.

CTRL-E support ?

No, not this way, mate !

Keyfile system

Yes, there's now a keyfile system used for this datatype - one could say, that this has been demanded, since it seems that most users obviously would like to get some value for their registration and also would like to see that "Registered ?" text disappear in the progressbar, after they indeed did register.

Please note, that the keyfile actually does not enable any "extra functionality" except making the PPC module fully functional and just replacing that "Registered ?" text in the progressbar.

Since the shareware fee of 15 DM is very low, and the keyfile is just an extra gimmick, 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 ! Otherwise I'd assume, that you don't need/want the keyfile...

If you registered the datatype earlier (when there wasn't a

keyfile system at all), simply send me an email and request your keyfile afterwards.

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 "akJFIF.wos" (150K) 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_WOSN 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 JPEG code will make use of virtual memory, which again will mean an even bigger slowdown. (*) 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 JFIF decoder module has to be placed at location SYS:Classes/Datatypes/akJFIF.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 (for 24 bit images, i.e. if not in V43 mode) is NOT yet PPC optimized - get a graphics card !

Please note, that this optional ELF decoder 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 the PPC module and make use of the plain 68k decoder
2. make use of the 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 JPEG 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 the 68k decoder, plus approximately another 110K 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.

Q: Why are there different speed-up factors for different images ?

I've performed Jan Uerpmann's PicBench test from his site <http://www.tu-bs.de/~y0002723/files/PicBench.lha> and it seems to indicate this.

A: The "larger" the images, the more the PPC can help increasing decoding speed; however, file size, image size and compression ratio of the JPEGs will influence the benchmark results, i.e. a small file with a high compression ratio may be more suitable for the PPC than a large file with only low compression (while keeping the image dimensions). Larger images, on the other hand may deliver better results than smaller images (keeping the compression factor constant). This benchmark does not check/proove this, we just tried "average" (accidental) images.

More datatypes ?

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

What's the relationship between JFIF and LJPG ?

The answer is simple: on file format level, there actually is none (*), thus neither one of the two datatypes does replace the other one and you would need to install both to get the functionality of both - but akJFIF perhaps is the more important one.

(*) well, there *is* a relationship, but it only does concern the file format construction and does not actually build a base for real upward/sideward compatibility

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.

Why are "progressive" image files not displayed progressively ?

Because picture.datatype's API (upto V43) relies on complete bitmaps to be returned by a datatype of subclass "picture".
Unfortunately the datatype cannot:

- supply many small bitmaps, one for each line
- give control back to picture.datatype during reading a file
- write into an existing, given bitmap

(to just supply some possible considerations how to solve this problem), so there currently is no way of displaying images progressively. Nevertheless the support of "progressive JPEG" is a noticeable feature, since (speaking of the JFIF file format) it is not backward-compatible with common JPEG and readers need to take special care for these files (JFIF V6a extensions).

When running in PPC mode, progressive display BTW would be a bad idea, anyway.

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.

IBrowse troubles

If you want to bypass the internal (68k) loaders of IBrowse and use the (PPC) datatypes instead, there may occur problems sometimes (not decoding and displaying all the graphics, but only some).

This seems to be caused by using the wrong priorities for internal and external decoders and data tranfers (and/or by MCP ;)

If you really want to use datatypes for decoding, you should set their priority to e.g. 10 and the others to 0.

(original report by Boris Bojic <bbogic@arco.met.fu-berlin.de>

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...

An other reason may be, that you did not delete all of the old (obsolete) "competing" JFIF descriptors from DEVS:Datatypes - in general, there should only remain JPEG/JFIF descriptor files that are shipped with the CURRENT release of this datatype.

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

Then, there are people who save images fromout a popular tool called Ph*t*sh*p (V4/5) as TIFF, give these the extension .jpg and wonder why these are not recognized as JPEG (whether these can or cannot be loaded by akTIFF, is a different story ;)

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://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 |
| | |
| | Andreas R. Kleinert |
| | Am Kornberg 48 |
| | D-57076 Siegen |
| | Germany, Europe |
| | |
| | +49-271-22869 |
| | (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 ask before sending binaries!
 And please think twice before asking - my postbox
 is not unlimited in size.

- Usenet
 >>> info@ar-kleinert.de
 Andreas_Kleinert@gmx.de
 Andreas_Kleinert@t-online.de

1.10 thanks

Thanks go to (in order of appearance ;-)

=====

- Martin Sprenger	- Kristian Phillips	- Swen K. Stullich
- Brad Avery	- Erik Magnusson	- Matthias Kraft
- Allan Odgaard	- Francesco Doro	- Per Jonsson
- Ariel Magnum	- Jürgen Klein	- Gabriele Guardì
- Arno Richter	- Philippe Reux	- Matteo Tenca
- Odd H. Sandvik	- Ingo Jürgensmann	- John Millington
- Jerome Lovy	- Carlos Lopez Otero	- Henk Jonas
- Albert Meyer	- Thomas Boerke	- André Laemmer
- Edwin H. Bielawski	- Matthias Schulze	- Dipl.Phys.Carl-Rudolph Naefe
- Trevor Daley	- Paul Wood	- Ludwig Berndt
- Roger Hågsen	- Dr. Rainer M. Herold	- Jan Skypala
- Thomas Nilsson	- Michael Schulz	- Thomas Steinbichler
- Roland May	- Jon B. Peterson	- M. A. Jones
- Andreas Mixich	- Robert Wahnsiedler	- Jörn Krüger
- Bodo Thevissen	- Helge Thorsten Kautz	- Troels Walsted Hansen
- Thomas Nolte	- Harry W. Turner II	- A. P. Suggitt
- Mat Bettinson	- Luco Nora Giorgio	- Dr. Greg Perry
- Stephen Bridges	- Philippe Duchenne	- Jure Dolanec
- Tom Lively	- Adam Atkinson	- Ben Vost
- Alexander Fichtner	- Dennis Lee Bieber	- Max Headroom
- Ian Barclay	- Marc-Tell Volkmann	- Christian Beck
- Torbjörn Aronsson	- Jürgen Haage	- Michael C. Battilana
- Jens Gössing	- Robert S. Puffer	- Dirk Busse
- Rolf Schuster	- Joel Alvim	- Christian Zimmermann
- Lothar Schülke	- Roland Gerecke	- Armin H. Pöhlmann
- Per Jonsson	- Les Morgan	- Roland Mainz
- Thorsten Ernst	- Wolfgang Wichmann	- Robert C. Reiswig
- Dave Sparks	- Uwe Cremerius	- Andreas Kramer
- Guillaume DuFour	- A J Price	- B & D Kubler
- Michael Aigner	- Christer Oldhoff	- Joel Alvim
- Frank Dietrich	- Donald Feldbruegge	- Arndt Bußmann
- Torsten Moll	- Georg Rottländer	- Phil Vedovatti
- Burkhard Breuer	- Ulrich Falke	- Aubert Pascal
- Martin Pape	- Sanjo Schiffmann	- Slobodan Todorovic
- Walter Gierholz	- Petra Struck	- Michael Steinke

- Bernd Mingers	- Wendell Watanabe	- Dr.-Ing. Heiko Pollmeier
- Ramiro Garcia	- Heiko Kröhnert	- Edward J. Barcik
- Rick Rudge	- Alvaro Thompson	- Achim Stegemann
- Bert Bosma	- Steve Brightman	- Michael Tobin
- Christian Hattemer	- Ignazzi Carmelo	- Eike Biel
- Heinz Rohner	- Christian Hattemer	- Kirk Strauser
- Dirk Hallen	- Jürgen Ofner	- Jürgen Barthmann
- Tilo Hanich	- Roman Patzner	- Klaus B. Küsche
- Jörg Handweg	- Stefan Michel	- Jochen Rhein
- David Newman	- Marco Vernaglione	- Bradley Rogers
- Simo Koivukoski	- Michael Jaccoud	- Jan Uerpmann
- David Gill	- Willi Demuth	- Sander Assenbroek Machielsens
- Achim Akkermann	- Steven Taylor	- Jörg Bierwagen
- Hans Eiblmeier	- Harry Miktarian	- Gerrit-kjeld Dusselje
- Yann Muller	- S.W. de Vries	- Gernod Schomberg
- Gerald Lorang	- Sebastian Becker	- Mario Kuchel
- Gabriele Greco	- Gérard Cornu	- Martin Mittelbach
- Hynek Schlawack	- Karl-Heinz Schulz	- Alexander Wissnet
- Anders Bolager	- Christian Hunyar	- Ralf Lillemäe
- Andreas C. Schmidt	- Daniel Kasmeroglu	- Frank Durban
- Gunnar Schuster	- Tim Pykett	- Thomas Körner
- Malcolm Harnden	- Christoph Kirsch	- Daniel Boerger
- Thorsten Marquardt	- Bjarke Vangsgaard	- Jukka Anttila-Vatjus
- Dave Fieldman	- Andrew Zalotocky	- Mark Carter
- Thomas Steffens	- Carsten Knodel	- Emmanuel Rey
- Sven Ottemann	- Matthias Laskowski	- Sebastian Abel
- Ralph Ewers	- Thomas Wiedecke	- D S Jeyasingh
- John Block	- John Jackson	- Robin Hüskes
- Vincent Morenas	- Neil Bothwick	- Javier Marcet
- Michael Merkel	- Ralph Ewers	- Steve Krueger
- Jim Cooper	- Clifford Mould	- Jon Steinar Kvaale
- Oliver Molz	- Klaus Müller	- John Aadnoy
- Sven Bornkessel	- Arvid Schlesinger	- Wolfgang Krause
- F. Ruthe	- Darin M. Weidenbach	- Alexander Niven-Jenkins
- Gary Goldberg	- Thomas Birk	- Vincenzo Morra
- Holger Kruse	- Swaraj Jeyasingh	- Michael Burkhardt
- Martin Potter	- Alan Surette	- Keith Blakemore-Noble
- Vincenzo Morra	- Ross Kirk	- Michel Verstraeten
- George Elliott	- Kevin Futter	- Michael Groni
- Markus Grubinger	- Kimme Utsi	- Andrew Baldwin
- Otto Carvalho	- Andreas Krüger	- Gerd Schniggenberg
- Luca Ricossa	- Phillip Wright	- Frédéric Faux
- Elmar Hoffmann	- Jonas Hultén	- Johann Samlowski
- Giambattista Comi	- Philippe Devilard	- Johan Eriksson
- Antonio Brianese	- Michael John	- Stefan Ohlsson
- Rune Jensen	- Jürgen Urbanek	- Mikkel Hald
- Hal Samuelson	- Norman Caetano	- Per Arne Flø
- Rich Robinson	- Adam Corrano	- Beth Hedrick
- Casper Thygesen	- Kai Foelster	- Peter Denomy
- Morten Straarup	- Thomas Karlsen	- Luca Baldelli
- Leonardo Petrucelli	- Richard Gore	- Tom Duin
- Dominique Deangili	- Anders Drejer	- Olivier Pertin
- Colin Keefe	- Roger Curtis	- Sam Gillies
- Linus Silvander	- Klaus-Peter Simon	- Whitford Bates
- Laurent Moussy	- Paul Kieffer	- Yves Liebercier
- S. Lichtendahl	- Alan Guillevic	- Keith Robertson-Turner
- Thomas Lorenz	- Chris Barrow	- Ed Eden

- Keith Schyler	- Trond K. Tveit	- Janko Köhler
- Andrew Mills	- Howard Toliver	- Jon Mines
- Ralf Mack	- Magnus Bouvin	- Dan Muldin
- Mahieux Pascal	- James Luscombe	- Peter Mattson
- Thomas Raukamp	- Steffen Häuser	- Kapryan Kennedy
- Coeurjoly Fabien	- William Eaves	- Cameron Snyder
- Martin Ruston	- Johnny Nielsen	- Jason Birnie
- Michael Osmolski	- Kevin Fairhurst	- Peter Annuss
- Sebastian Eichholz	- Larry Urquhart	- Philip Yearbury
- Neil Bowes	- Steve Hodson	- Johan Rönblom
- Harald Schulz	- Christian Schröpfer	- Michael Fedrowitz
- Denis Zwornarz	- Gert Hubers	- Robert Little
- Christopher Handley	- Stefano Guidetti	- Jürgen Seubert
- Paul Korhonen	- Frank Müller	- Peter Kaltstein
- Sebastian Cramer	- Peter Theuring	- Gunter Kusserow
- Telemar Rosenberger	- Phillip Degnan	- Alexandre Kairouannais
- Jörg Dreier	- John Melville	- Chris Dallimore
- Paul Sadlik	- Matthew Sawyer	- Jeffrey Grzanich
- John Hart	- Ian Tyrell	- Walt Challender
- Martin Sprenger	- Dr. Rüdiger Kielmann	- Pekka Sippola
- Brice Terzaghi	- Adrian Cope	- Frank Böhne
- Petr Voralek	- Antoine Bordier	- Patrice Dumont
- Manfred Kern	- Francis Klein	- Dominique Harelle
- Arnljot Arntsen	- Havard Lunde	- Geoff Tovey
- Herve Sonnevile	- Winek Zawada	- Sascha Ploss
- Paul Lang	- David Gerber	- Michael Domoney
- Carl Read	- Harald Wuensche	- Eirik Synnes
- James Harrison	- Mark Shaw	- Frank Wille
- Adam Suwala	- Winfried Krueger	- Simon J Glover
- Stephan Neise	- Wolfgang W. Wolber	- Don Cox
- Henrik Jensen	- Matteo Consolati	- Jürgen Wilschke
- Stephen Webber	- Clive Dennett-Thorpe	- Svein Inge Wik
- Philippe Reux	- Paul Venton	- Bjarke Vangsgaard
- Rolf Max Rotvel	- Michael Every	- Jason Ruellan
- Stefan Fischer	- Lars Henrik Jensen	- Roberto Muller
- Michael Thompson	- Alfred Kendall	- John Orwin
- Holger Beer	- Mike Weling	- Rolf Kleiber
- Kari Kortro	- G. Burdett	- Daniel Westerberg
- Daniel Stripes	- Scott Konowal	- Steinar Pedersen
- William Seaton	- Brian Riis	- Antonio Maria Sebastiani
- Dario Soccoli	- Arno Richter	- Chris Appleton
- Richard Lane	- Manfred Kern	- Marco van der Laan
- Christian Sauer	- Rasmus Bothe	- Neil Mallet
- Andreas Ohlsson	- Mark Vallins	- Paul Compton
- Craig Peterson	- Gontier Laurent	- Simon Jones
- Mathias Roslund	- John de Boni	- Maria Pelova
- Jennifer Symancyk	- David Hibbert	- Bruno Caruso
- Wolfgang Bauer	- Michael R. Wilson	- Arsi Koutaniemi
- Arthur Moyer	- Thomas Schaefer	- James Miller
- Karl-Olav Gravdahl	- Janifer Lopez	- James George
- Eric Muller	- Ian Argæet	- Mats-Olov Rustad
- Ian Armstrong	- Philip Vedovatti	- Daniel Plant
- Christian Kersting	- Klaus-Dieter Klang	- Alexander Schröder
- Dirk Pohlmann	- Josef Mayr	- Dietbert Leusmann
- Markus Schmidt	- Frank Aisenbrey	- Kevin Hupp

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

akJFIFPrefs

akJFIFPrefs is the Preferences Program for akJFIF.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.

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/akJFIF.prefs" to an optional task-related prefs file called

"Datatypes/akJFIF.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/akJFIF.prefs_Tasks
```

```
Copy ENV:Datatypes/akJFIF.prefs ENV:Datatypes/akJFIF.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) DECODE_METHOD=(FAST_INTEGER|SLOW_INTEGER|FLOATING_POINT)
- 2) DITHERMODE=(NO_DITHERING|DITHER_FLOYD-STEINBERG|DITHER_ORDERED|HAM_OUTPUT)
- 3) V44_DITHER=(0..2)
- 4) V43MODE=(256|24BIT)
- 5) V40_DEPTH=(3..8)
- 6) V40_COLORSPACE=(COLOR|GRAY)
- 7) HAM_MODE=(HAM6|HAM8)
- 8) COLOR_QUANTIZING=(SLOW|FAST)
- 9) UPSAMPLING=(ON|OFF)
- 10) SCALE_DENOM=(0..3)
- 11) INTERLEAVED_BM8=(ON|OFF)
- 12) DISPLAYABLE_BM8=(ON|OFF)
- 13) PROGRESSBAR=(ON|OFF)
- 14) CUSTOM_MODES
- 15) PPC=(ON|OFF)
- 16) AUTO=(ON|OFF)
- 17) PPCLIB_EMU=(IGNORE|USE)
- 18) CACHEWOS=(ON|OFF)
- 19) LOADELF_WOS=(ON|OFF)
- 20) NOASPECT
- 21) DEBUG

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

```
V44_DITHER=1
DECODE_METHOD=FAST_INTEGER
DITHERMODE=DITHER_ORDERED
V43_MODE=24BIT
V40_DEPTH=8
V40_COLORSPACE=COLOR
HAM_MODE=HAM6
COLOR_QUANTIZING=FAST
UPSAMPLING=OFF
PROGRESSBAR=ON
SCALE_DENOM=0
INTERLEAVED_BM8=ON
DISPLAYABLE_BM8=OFF
AUTO=ON
PPCLIB_EMU=IGNORE
CACHE_WOS=ON
LOADELF_WOS=ON
```

General Explanation of Options

=====

1) DECODE_METHOD

```
SLOW_INTEGER:    slow, but accurate integer algorithm
FAST_INTEGER:    faster, but less accurate integer algorithm
FLOATING_POINT:  floating point algorithm
```

2) DITHERMODE

NO_DITHERING: no dithering, best pen selection
DITHER_FLOYD-STEINBERG: slow, high quality, floyd-steinberg dithering
DITHER_ORDERED: ordered dithering to fixed palette
HAM_OUTPUT: generate HAM output from 24 Bit data

The datatype will only output 24 Bit data, when
DITHERMODE=NO_DITHERING _and_ V43_MODE=24BIT are set.
Otherwise even with V43 picture.datatype the data will
be dithered, HAM-converted or best-pen colormapped.
Thus DITHERMODE=NO_DITHERING activates the V43_MODE switch
for picture.datatype V43+

3) V43_MODE (only with "NO_DITHERING" and picture.datatype V43+)

256: disable 24 Bit output even for V43 picture.datatype
24BIT: do raw 24 Bit output with V43 picture.datatype

The datatype will only output 24 Bit data, when
DITHERMODE=NO_DITHERING _and_ V43_MODE=24BIT are set.
Otherwise even with V43 picture.datatype the data will
be dithered, HAM-converted or best-pen colormapped.
Thus V43_MODE will only act as a switch, when DITHERMODE=NO_DITHERING
has been set and picture.datatype V43 is running.

4) V40_DEPTH

When dithering to a palette (so: when in V40 mode) 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}).

5) V40_COLORSPACE

Whether you want to get 16..256 colors or 16..256 grayscales when
requesting V40_DEPTH.

6) HAM_MODE (only with "HAM_OUTPUT")

HAM6: generate HAM6 output (max. 4096 different colors on screen)
HAM8: generate HAM8 output (262144+ different colors on screen)

When in HAM8 mode, gray images will be exported as 256 color
gray images nevertheless (they do look ugly in HAM8 and AGA
machines CAN handle 256 colors).

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.

7) COLOR_QUANTIZING (with "DITHER_FLOYD-STEINBERG" only)

FAST: high speed color quantization
SLOW: high quality color quantization

8) UPSAMPLING

ON: do careful upsampling of chroma components
 OFF: do faster, but sloppier upsampling of chroma components
 (often very small visual impact)

9) SCALE_DENOM

0..3: use a scale factor of either 1/1 (none), 1/2, 1/4 or 1/8
 for graphics reading.
 For reasons of better failure safety checks, this is specified
 as an exponent value between 0 and 3 ($2^0=1$, $2^1=2$, $2^2=4$, $2^3=8$).

10) 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. EGSPlus

11) 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.

12) PROGRESSBAR

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

13) 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.

14) 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.

15) 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).

16) 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+

17) CACHE_WOS

This option is explained in the FAQ.

18) 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).

19) 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.

20) DEBUG (hidden option)

Not implemented yet. Will enable debugging output, i.e. info requesters.

Explanation of settings
=====

Please note, that it does not suffice to simply specify the preferred options. Some also do react as switches. Here are some examples how to manage specific configurations:

```

HAM Output (V40/V43):      DECODE_METHOD=FAST_INTEGER      *1
                           DITHERMODE=HAM_OUTPUT           *1
                           V43_MODE=256                   x1
                           HAM_MODE=HAM8                  *1
                           UPSAMPLING=OFF                 *1

Dithered Output (V40/V43): DECODE_METHOD=FAST_INTEGER      *1
                           DITHERMODE=DITHER_ORDERED      *2
                           V43_MODE=256                   x1
                           COLOR_QUANTIZING=FAST           *1
                           UPSAMPLING=OFF                 *1

256 Colors (V40/V43):      DECODE_METHOD=FAST_INTEGER      *1
(Best Pen selection)       DITHERMODE=NO_DITHERING         *1
                           V43_MODE=256                   x2
                           COLOR_QUANTIZING=FAST           *1
                           UPSAMPLING=OFF                 *1

24 Bit (V43):              DECODE_METHOD=FAST_INTEGER      *1
                           DITHERMODE=NO_DITHERING        *1
                           V43_MODE=24BIT                  x2
                           UPSAMPLING=OFF                 *1

```

```

x1 here: always ignored
    (because DECODE_METHOD not equal NO_DITHERING)
x2 here: acts as a switch between V40 and V43 mode,
    when running picture.datatype V43
    (because DECODE_METHOD=NO_DITHERING)
*1 may be changed as preferred
*2 may be changed to: DITHER_FLOYD-STEINBERG
    (side-effects described by "x1" and "x2")

```

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'

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

=====

- V44.112 (14.8.2001): - removed progress-bar support (for size + speed)
- optimizations
- no longer using utility.library
- V44.98 (5.4.2001): - did I mention, that I only believe in benchmarks
that I did manipulate myself (freely after Churchill,
on "statistics") ?
- V44.97 (24.3.2001): - ...
- fixed the long-standing exit-on-error bug,
with stackswap not being reverted (-> Laire)
- V44.96 (16.3.2001): - V44.95 did not get installed due to wrong \$VER: tag
- speedup(s) for 68k and PPC
- V44.95 (10.2.2001): - speedup(s) for 68k and PPC
- removed support for WOS crap; use ppplibemu instead...
- V44.90 (3.2.2001): - minor speedup
- V44.82 (8.8.2000): - SPEEDUP option disabled (breaks things now)
- V44.81 (7.8.2000): - general speed up for all three versions (68k/PPC/WOS),
especially on smaller files
- slight speedup of unregistered versions
- speedup of registered version
- V44.79 (13.5.2000): - added support for another variant of JFIF-EXIF files
(to make file recognition more error-proof, a new
JFIF-EXIF2 descriptor has been added instead of
making the old one more "general"). This allows
for support of more digital cameras now.
(-> Neil Bothwick)
- added german translation

- Aminet release
-

V44.78 (24.3.2000): - updated docs, etc.

V44.77 (2.2.2000): - 68k: fixed "out of memory" bug with conventional IO,
i.e. when `asyncio.library` was not present
(-> Colin J. Knight)

V44.76 (19.1.2000): - added new Installer script by Robert C. Reiswig

- Aminet release

V44.75 (7.1.2000): - fixed revision history of V44.70

- WOS: - memory pools (`powerpc.library` V15+)
- PPC: - fixed small bug
- if the standard JPEG 68k datatype of OS 3.5 does not fulfil your needs, then please REGISTER this one...
- 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 wich 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!
- SView Productivite Suite II CD-ROM by Schatztruhe (Germany) or Software Hut (US) does include full versions of:
 - SViewIV
 - akJFIF, akPNG, akTIFF and akNAIL datatypes
 - akMPEG2, akMPEG4

Plus a lot of extras!

- for the newest version of akMPEG2/4, please take a look at www.ar-kleinert.de ...

V44.70 (5.1.2000): - 68k: - making use of `asyncio.library` when available, but own IO routines in any case now (V40.65)
- fixed bug in memory pool handling; it seems to have worked flawlessly only due to a lucky side-effect
- made new IO routines slightly smarter
- non-async part of custom IO was slower than before;

fixed now

- PPC: - fixed bug in memory pool handling; it seems to have worked flawlessly only due to a lucky side-effect

- WOS: - added PPC603-optimized version

V44.65 (4.1.2000): - PPC: - faster code
- making use of memory pools now (libjpeg only)
- optimized memory allocation

- WOS: - optimized memory allocation

- 68k: - memory pools
- optimized code
- making use of `asyncio.library` when available

- Note: memory optimizations especially should show effects on PPC-603e systems with low memory bandwidth.

- added OS 3.5 dithering control preferences option

V44.60 (16.12.99): - added a few additional notes about OS 3.5 compatibility to FAQ (forget Usenet information - I'm not writing there anymore, either ;-))

- added SWIFT code information for online money transfer

- split archives into 68k, PPC and WOS parts

- Aminet release

V44.50 (22.11.99): - SPEEDUP option now limited to pic-dt V43+ and slightly changed - still only works with AWeb (AWebIP) task-specific settings files and heavily depends on how AWeb deals with Bitmaps

- tried to make JFIF error handling more verbose, regarding insufficient memory or decoding errors (68k/PPC/ ↵ WOS)

- fixed possible memory leaks on error, a smaller one and a big one (PPC)

- Aminet release

V44.49 (31.10.99): - OS 3.5 cleanup: - added note about "16 bit dithering" problem to FAQ
- updated docs

V44.48 (22.10.99): - Aminet release

- maintenance release

V44.47 (22.09.99): - PPC/WOS: - decoder speedup (DCT)

- address change
 - Aminet release
 - V44.46 (25.08.99): - fixed docs and forms
 - Aminet release
 - V44.45 (10.08.99): - prefs GUI was broken
(-> Christian Sauer)
 - datatype still is V44.44
 - V44.44 (08.08.99): - added new installer script by Rob Reiswig
 - prefs GUI fixes
(-> Georg Rottlaender)
 - V44.43 (18.07.99): - my postal address will change. Updated docs!
 - progress bar's position and title text changed
(-> Petr Voralek)
 - Aminet release (25.07.99)
 - 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
 - V44.41 (04.07.99): - gray JPEGs would have been scaled to 1/3 of
their width when being displayed in HAM6/8;
or something like that. Fixed now. [68k only]
(-> Michal Zubrzycki)
 - gray JPEGs now always will be exported as
256 color images, even in HAM8 mode
(however, not in HAM6 mode) [68k & PPC]
(-> Michal Zubrzycki)
 - V44.40 (26.06.99): - JFIF-ADOB now fits for more image types
(-> Phil Wright)
 - 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
 - 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
- added new descriptor called "JFIF-ADOB" for JPEG files that had not been recognized before; these do contain the string "Adobe" which seems to indicate that they're written by an old version of Photoshop or an other graphics program by this company; only had two images for testing...
- 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): - "DEBUG" now is a reserved, hidden switch (non-functional yet)
- 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
- V44.24 (12.02.99): - DANGER: LOADELF_WOS option now is a switch, either speficy 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: small changes
- V44.19 (07.01.99): - version bump
- 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)
- V44.15 (01.01.99): - new-year cleanup
- forgot to bump version last time (still was 44.13)
- fixed SCALE_DENOM bug, which caused the image to be scaled, but not the bitmap (same size, but with black area). PPC/WOS only.
(-> Pascal Mahieux)
- it's now checked whether akJFIF.wos is available, before it's tried to open powerpc.library and also it's now checked for akJFIF.ppc before ppc.library is opened - this way you can disable either PPC support by simply deleting the corresponding file; this may be useful for people who normally use PPC-Lib but have powerpc.library V14 installed.
(-> Jürgen Urbanek)
- fixed small bug in WOS part (settings handling)
- V44.14 (22.12.98): - fixed documentation at various places
- V44.13 (20.12.98): - ELF loader for WOS could have crashed under low memory conditions (did not check correctly for AllocVec return value, once)
- completely rewrote ElfLoader; added WOSLoadObject(), WOSUnLoadObject(), WOSCreateTask() functions, in analogy to ppc.library
- added CACHE_WOS option; this one will avoid re-loading of the ELF module from disk but instead keep it in memory (needs twice as much memory, even during the decoding process, but should be noticeably faster)
- V44.12 (19.12.98): - fixed some typos in the docs and related text files
- updated docs
- added PPaint 'stack crash' note
- prefs program now again compiled using SAS/C instead of StormC: reduced size from 49440 bytes to 29520
- this also fixes possible stack problems with the prefs
- added new installer by Robert C. Reiswig (taking
-

- care of any possible .wos and LoadElfWos files)
 - added akJFIF.wos plugin for optional WarpUp/WarpOS support (leave akJFIF.ppc where it is - either one will be used automatically)
 - 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)
- V44.6 (16.9.98):
- added modified Prefs GUI by Georg Rottlaender
 - the newest wizard.library version seems to be V41.101; you can find it in the archive with the AmigaWriter demo version under ftp.haage-partner.com; however, if the prefs still tend to crash, maybe you'd just need to adjust the stacksize to 32768 bytes...
- V44.5 (1.9.98):
- now at least requires ppc.library V45 (V46 recommended !)
 - 68k I/O speedup
 - general speed improvements (68k)
 - prefs program now allows loading of task-specific (or again default) settings files (via menu); saving already was possible
 - PPC: under OS 3.1 pre-dithered/remapped JFIFs (V40 mode or best-pen) with upto 256 colors now will be moved faster to their destination bitmaps (registered version, only).
 - removed history entries for versions below 44.1
- V44.4 (9.8.98):
- *** MAJOR RELEASE ***
- This version seems to be quite stable now.
Updates will appear when necessary.
- stripped ELF module (~2000 bytes)
- V44.3 (29.7.98):
- semaphore locking now more restrictive; possible problem when under heavy parallel use of the PPC decoder ?
 - stackswap in LibInit now only when necessary and only to 8K (not 16K)
 - rewrote the docs section about 040/060 and math patch recommendations; don't patch your system worse !
 - addressed minor cacheflush problem (strlen+1)
 - rewrote major parts of the documentation; check it out!
 - PPC: fixed problem with partially trashed 24 bit bitmaps when in demo mode; assumed that the destination bitmap was empty (== black lines), which wasn't necessarily true. Now explicitly clearing those lines, which MIGHT slow down the PPC demo mode when compared to the registered PPC mode.
 - fixed "can't close shell window problem"
-

- (thanks to Michael Merkel for pointing out)
- fixed problem with SAS/C's stdio initialization that could cause crashes when there was tried to do an `Open("*, ...)` - now all the three stdio handles are NIL: since we don't use these, anyway. This finally should fix the problems we sometimes ran into with DOpus and IPrefs/WBPattern. Also, they're now properly unlocked (which was a problem related to the reuse of our ELF module and caused those shell problems).
- (-> thanks to Michael Merkel, Javier Marcet and Ralph Ewers for beta testing, Olaf Barthel, Frank Mariak, Dr. Greg Perry and Jonathan Potter for useful comments and Steve Krueger and Jim Cooper for all their work on SAS/C for PPC as we have it now).
- fixed 1032 byte memory leak, as introduced in one of the latest versions
- (-> Troels Walsted Hansen)
- fixed problem in prefs file handling, that could occur under low memory conditions
- PPC: the demo version wasn't actually displaying every third line, but... following a different scheme
- (-> Javier Marcet)

- V44.2 (15.7.98):
- added info on new, permanent Shareware discount (bundling) offer
 - added info on FBlit, FastBlit, CpuBlit98 to FAQ (get your bitmaps into fast ram and/or utilize the CPU for blitting)
 - increased stacksize in prefs' icon from 4096 to 32768 bytes to avoid crashes from WB
 - added IBrowse info to FAQ
 - made some efforts to reduce stack usage within the datatypes' 68k code where possible; may help to avoid crashes sometimes
 - iffparselibrary was opened although not needed
 - added some missing IoErr()s
 - added temporarily 16K stackswap to init code, where it is safe (I will `_not_` do that for the main datatype dispatcher, it would most likely cause a bunch of new problems)
 - added StackCheck mechanism that will put up a requester and will allow you to make the choice to quit - giving a low memory error - before its too late. It's been tested with MultiView and DOpus5.
 - if you get "ramlib" gurus with this or any other program, then try installing the following patch:
Aminet:util/sys/StackAid.lha

- V44.1 (27.6.98):
- prefs program now allows to do task-specific settings directly
 - jumped to V44 since some people seem to prefer version inflation over clarification

- you should upgrade to ppc.library V46 (ftp.phase5.de)

1.13 About PNG - successor of GIF

PNG is the successor of the GIF file format - it's one of the official graphic file formats that are recommended by the W3C for use in the WWW. Other than GIF it is completely free of patent claims and has been designed with free data exchange in mind. Drop GIF for PNG - free algorithms are as important as free speech on the internet:

GIF is obsolete - you neither should use nor support it any longer. If you are doing WWW design, use PNG and JPEG instead. It's important !

For more information on PNG (pronounce: PiNG) for example look at:

- [1] PNG specification (AmigaGuide format)
-> Aminet:docs/hyper/PNG-guide.lha
- [2] PNG WWW homepage
-> <http://www.cdrom.com/pub/png/>
- [3] PNG upgrade tools like gif2png
-> Aminet:gfx/conv/gif2png-0.6.lha
- [4] programs capable of PNG, like PPaint, SuperView,
or PNG-Box, etc.

1.14 PNG-Box - WWW tool for PNG writing

PNG-Box

- SHAREWARE -

© 1997-2000 by Andreas Ralph Kleinert. All rights reserved.

A PerSuaSiVe SoftWorX PRODUCT.

Program information

Now you can easily switch to PNG !

PNG-Box loads graphics files via SuperView-Library and allows to convert these to PNG (PiNG) file format for WWW usage with several WWW-specific options to be set:

- progression on/off
- transparency on/off
(and set a transparent color ranged in 0..maxcolors)
- compression 0..9

The GUI will show you compression efficiency (byte sizes) and display various other useful information.
It's style guide conformeous and based on wizard.library.

It's also possible to view the PNG compression source and destination files in a window on workbench. This way there's a preview possible (since PNG is non-lossy, it always should look similarly, though ;)

Includes PPC/WOS optimized saver module (registered version, only) and takes advantage of SuperView-Libraries' PPC/WOS (loader) modules as well.

See program archive for copyright and distribution information.
See Aminet:gfx/conv/PNG-Box.LHA for download.

1.15 JPEG-Box - WWW tool for JPEG writing

JPEG-Box

- SHAREWARE -

© 1998-2000 by Andreas Ralph Kleinert. All rights reserved.

A PerSuaSiVe SoftWorX PRODUCT.

Program information

Now you can easily convert your graphics to (progressive) JPEG (JFIF) !

JPEG-Box loads graphics files via SuperView-Library and allows to convert these to JPEG (JFIF) file format for WWW usage with several WWW-specific options to be set:

- progression on/off
- compression 0..100 (attention: "0" means BIGGEST compression !)
- (various other settings)

The GUI will show you compression efficiency (byte sizes) and display various other useful information.
It's style guide conformeous and based on wizard.library.

It's also possible to view the JPEG compression source and destination files in a window on workbench. This way there's a preview possible.

Includes PPC/WOS optimized saver module (registered version, only) and takes advantage of SuperView-Libraries' PPC/WOS (loader) modules as well.

See program archive for copyright and distribution information.
See Aminet:gfx/conv/JPEG-Box.LHA for download.
