

akPNG_Documentation

COLLABORATORS

	<i>TITLE :</i> akPNG_Documentation		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		August 22, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	akPNG_Documentation	1
1.1	Commercial Break	1
1.2	akPNG : Documentation	1
1.3	copyright	3
1.4	disclaimer	3
1.5	distribution	4
1.6	payment	4
1.7	Usage and so on	4
1.8	Datatype FAQ	5
1.9	Making use of 680x0 CPUs and PPC accelerators	10
1.10	correspondence	12
1.11	thanks	13
1.12	prefs	15
1.13	history	17
1.14	About PNG - successor of GIF	29
1.15	PNG-Box - WWW tool for PNG writing	29

Chapter 1

akPNG_Documentation

1.1 Commercial Break

DON'T click here > < or here > <

```

-----
*****  *****  *****  *****  ++++++
*          *      *      *      *      *      +   +
*****    *      *      *      *****  +++
          *      *      *      *
*****    *      *****  *              +
-----

```

and now to something completely different...

<Commercial> BTW: What is SViewNG ? </Commercial> with PPC support !

<Commercial> Already tested PMPPro ? </Commercial> with PPC support !

...before you E N T E R the DATATYPE documentation!

(no this isn't an aggressive commercial - if you want to see one,
just wait for the NEXT release ;-)

1.2 akPNG : Documentation

akPNG.datatype V43.187

- SHAREWARE -

© 1996-98 by Andreas Ralph Kleinert. All rights reserved.

A PerSuaSiVe SoftWorX PRODUCT.

Needs Kickstart V3.x

Release Date : 5.6.1998

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

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

<Commercial> Already tested PMPPro ? </Commercial>

Copyright
Disclaimer
Distribution
Payment
Usage and Notes
Free algorithms... PNG: successor of GIF ...and free speech !
Datatype FAQ
68020-68060, PPC
Prefs
Correspondence
Thanks
Version-History

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

Please visit:

WWW Support Site
<http://www.amigaworld.com/support/akpng/> (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.

1.3 copyright

The akPNG.datatype in this version and its documentation files are (C)opyright 1996-98 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 to the author.

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

libpng:

libpng 1.0.1

Copyright (c) 1995, 1996 Guy Eric Schlnat, Group 42, Inc.

Copyright (c) 1996, 1997 Andreas Dilger

Copyright (c) 1998 Glenn Randers-Pehrson

zlib:

zlib 1.1.2

(C) 1995-1998 Jean-loup Gailly and Mark Adler

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

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.

1.4 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.

1.5 distribution

The akPNG.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.

1.6 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).

1.7 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 akPNGPrefs command to SYS:Prefs/Datatypes (optionally).

The .ppc module (manually) has to be placed into the same directory where the .datatype file goes to (usually SYS:Classes/Datatypes).

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 akJFIF-dt).

If there's enough RAM available, VMEM: won't be used.

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

akPNG.datatype is a new PNG datatype, which is based on the latest PNG sources (zlib V1.1.2, libpng 1.0.1).

So it does support 8 Bit color mapped files (colorspace expanded to 8 bit per component always) and True color files (24/48 Bit, alpha channel ignored, 48 Bit 16:16:16 cut down to 24 Bit 8:8:8).

So the following types of PNG images (all valid ones) should be imported in the described way:

Bit depths	Interpretation
1,2,4,8,16	pixels are grayscaled samples
8,16	pixels are R,G,B triple samples
1,2,4,8	pixels are palette indices

(plus variations with - here ignored - alpha channel)

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).

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

1.8 Datatype FAQ

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 (ELF)

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

For this, the ELF PNG decoder module has to be placed at location SYS:Classes/Datatypes/akPNG.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 this optional ELF decoder only will become fully functional for registered users of this datatype, which 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 PNG 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.

NOTE: decoding will need about twice as much memory as with

the 68k decoder, plus approximately another 145K 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 version 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 PNGs 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 akJFIF, akLJPG, akSVG and the co-production FAXX (with GPSOFT) datatype.

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.

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
```

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 "interlaced" 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.

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

Transparency (general)

PNG supports transparency levels for each color out of a given image.

For colormapped images, this is managed via a "shadow" colormap,
which supplies 0..255 ranged values for specific colors:
"0" means "fully transparent", while "255" means "not transparent at all".

Since the `datatypes` interface (upto V43) is not prepared for handling such cases, we simply search for the first transparent color, which matches a transparency level of "0", thus semi-transparent colors are ignored always (you can't reproduce these on a 256 color display, anyway). So, if you create your own WWW pages containing transparent PNG graphics: please make sure that there's only one transparent color being used, and that this one actually is fully transparent! The number of the transparent color is irrelevant - many people prefer color #0, though.

Transparency for true/high color images (more than 256 colors), i.e. via an alpha channel, is not supported at all (and perhaps never will be, for the V43 `picture.datatype`).

Transparency (esp. Browsers)

There have been many bug reports, where people told me, that the transparency features (you know, many web pages do contain "PNG" graphics with one color being transparent, thus just equal to the background color) did not work at all. All I can say about is, that at my current state of information this is not my fault. The transparency information as such definitely is being read correctly, and there is only one necessary step to be done - it needs to be passed to `picture.datatype` by setting a special flag in the `BitMapHeader` structure:

```
bmhd->bmh_Masking      = mskHasTransparentColor;
bmhd->bmh_Transparent = (UWORD) ((WORD)trans);
```

With `pic-dt` V43 there once also was a (now obsolete) special flag for that, but we don't use it (tested it, though):

```
PDTA_TransRemapPen, (LONG)bmhd->bmh_Transparent
```

It seems, that neither `pic-dt` V39/40 nor V43 do interpret that flag correctly in neither mode (with remapping or without).

Theoretically, there are two possible ways for a program (e.g. a browser) to handle a `datatype` graphics:

```
let picture.datatype do it
-----
- load it
- attach it to a screen/window and tell it where to appear
  in which size; allow remapping to the screen's colors
=> in this case, picture.datatype would have to manage the
    transparency handling and replace the transparent
    color's colormap entry with the corresponding screen color's
    values BEFORE remapping to the screen.
    It's SUBJECT TO THE PICTURE-DATATYPE.

do it yourself
-----
- load it
- get it without remapping
- remap and display it by yourself, also handle
```

```

    transparency by yourself
=> thus transparency won't be handled by the datatype at all.
    It's SUBJECT TO THE BROWSER.

```

Obviously both ways don't work with the current release, although I've been told, that an other datatype does the job correctly. Funny enough, the author did tell me, that he did program it the same way as I did.

Well, all I can say is: send any further bug-reports plus the explanation above to your Browser vendor or Pic-Datatype supplier. Can't do anything more about that, until someone tells me, where my assumptions are wrong (but I am not going to screw up the OOP datatypes concept just because of that and do the remapping just rightly to a possibly given screen by myself).

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.

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...`

1.9 Making use of 680x0 CPUs and PPC accelerators

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

But 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.

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...
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.
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". Don't install ppc.library without having a PPC board plugged in.

4. Read the corresponding FAQ pages for more information on PPC support and configuration.

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's a solution: it is strongly suggested, to use some of the available math-library replacements and/or patches from Aminet.

1. FMath by Martin Berndt, for example does replace all of them:
 - mathieeedoubbas.library
 - mathieeedoubtrans.library
 - mathieeesingtrans.library
 - mathtrans.library

It is strongly recommended to install V40.6 or a newer version from Aminet (Aminet:util/libs/FMath406.LHA).

2. FFPpatch by Jess Sosnoski additionally patches some functions of mathffp.library to use 68881/2 instructions. It currently makes use of some mathtrans.library functions for FFP/IEEE conversion and thus does require FMath as well. (There's a 'ffptest' program included, so that you can test, whether it improves speed or makes it worse.)

It is strongly recommended to install V1.7beta or a newer version from Aminet (Aminet:util/boot/ffppatch.lha).

On a A4000 with A3640 (68040-25) running FMath, it gave the following results:

[before]

mathffp.library speedtest
© 1997 Jess Sosnoski

Test Name	Time in Seconds

SPAbs()	took: 0.79 sec
SPNeg()	took: 0.80 sec
SPAdd()	took: 1.38 sec
SPSub()	took: 1.31 sec
SPMul()	took: 1.87 sec
SPDiv()	took: 2.05 sec
SpFlt()	took: 0.76 sec

```
SpFix()    took:    1.04 sec
SpFloor()  took:    1.09 sec
SpCeil()   took:    1.99 sec
```

[after]

```
mathffp.library speedtest
© 1997 Jess Sosnoski
```

Test Name	Time in Seconds
SPAbs()	took: 0.52 sec
SPNeg()	took: 0.48 sec
SPAdd()	took: 1.06 sec
SPSub()	took: 1.05 sec
SPMul()	took: 1.06 sec
SPDiv()	took: 1.32 sec
SpFlt()	took: 0.49 sec
SpFix()	took: 0.77 sec
SpFloor()	took: 0.82 sec
SpCeil()	took: 1.73 sec

- 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:

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

- 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 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.

It is strongly recommended to install the newest version from Aminet (Aminet:util/boot/Mult64Patch.lha).

If you do own a CyberStorm 060 board, just ignore what I said before and follow the instructions given in the 68060-Library documentation.

1.10 correspondence

WWW Support Site
<http://www.amigaworld.com/support/akpng/> (AWeb-II)

PNG Homepage: <http://www.cdrom.com/pub/png/>

```

|   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   |
|   Sandstrasse 1   |
|   D-57072 Siegen   |
|   Germany, Europe   |
|
| Any snail mail to the old address will still be routed. |
|
|   Phone:  +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.   |
|-----|

```

E-Mail:

Please send binaries via ARK@News.wwbnet.de, and keep them smaller than 16 KB. 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
          ARK@superview.ftn.neckar-alb.de   (Fido-Gate)
          Andreas_Kleinert@t-online.de      (T-Online)
          ARK@News.wwbnet.de                (Z-Netz)
          ARK@amigaworld.com                (AmigaWorld)

```

- 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

1.11 thanks

Thanks go to (in order of appearance ;-)

=====

(some of these people did register, others did make suggestions/bug reports or helped otherwise - how about you ?)

- Ingo Jürgensmann	- Thomas Boerker	- Andreas Mixich
- Robert Wahnsiedler	- André Laemmer	- Edwin H. Bielowski
- Matteo Tenca	- Jan Skypala	- Adrian Demarais
- Ludwig Berndt	- Roger Hågensen	- Dipl.Phys.Carl-Rudolph Naefe
- Dr. Rainer M. Herold	- Thomas Steinbichler	- Jörn Krüger
- Bodo Thevissen	- Helge Thorsten Kautz	- Thomas Nolte
- Harry W. Turner II	- A. P. Suggitt	- Mat Bettinson
- Vulture	- Dr. Greg Perry	- Stephen Bridges
- Philippe Duchenne	- Jure Dolanec	- Tom Lively
- Alexander Fichtner	- Magnus Holmgren	- Max Headroom
- Ian Barclay	- Marc-Tell Volkmann	- Christian Beck
- Torbjörn Aronsson	- Jürgen Haage	- Michael C. Battilana
- Milco Veljanoski	- Robert S. Puffer	- Jérôme Lovy
- Dirk Busse	- Armin H. Pöhlmann	- Karl-Heinz Ostertag
- Joel Alvim	- Per Jonsson	- Les Morgan
- Roland Mainz	- Robert C. Reiswig	- Dave Sparks
- Andreas Kramer	- Guillaume DuFour	- A J Price
- Michael Schulz	- B & D Kubler	- Christer Oldhoff
- Arndt Bußmann	- Torsten Moll	- Georg Rottlaender
- Phil Vedovatti	- Burkhard Breuer	- Ulrich Falke
- 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
- Alvaro Thompson	- Achim Stegemann	- Bert Bosma
- Ignazzi Carmelo	- Eike Biel	- Heinz Rohner
- Frank Dietrich	- Kirk Strauser	- Dirk Hallen
- Tilo Hanich	- Roman Patzner	- Klaus B. Künsche
- Jörg Handweg	- Stefan Michel	- Jochen Rhein
- David Newman	- Bradley Rogers	- Simo Koivukoski
- Michael Jaccoud	- Jan Uerpmann	- Achim Akkermann
- David Gill	- Willi Demuth	- Sander Assenbroek Machielsens
- John Millington	- Jörg Bierwagen	- S.W. de Vries
- Hans Eiblmeier	- Yann Muller	- Gerrit-kjeld Dusselje
- Gernod Schomberg	- Gerald Lorang	- Sebastian Becker
- Mario Kuchel	- Gérard Cornu	- Martin Mittelbach
- Karl-Heinz Schulz	- Anders Bolager	- Christian Hunyar
- Ralf Lillemäe	- Andreas C. Schmidt	- Daniel Kasmeroglu
- Frank Durban	- Gunnar Schuster	- Thomas Körner
- Malcolm Harnden	- Christoph Kirsch	- Jukka Anttila-Vatjus
- Thorsten Marquardt	- Rudy van Merkom	- Tristan R. Young
- Niko Tomatsidis	- Hans Flüss	- Pierre Radestock
- Michael Thompson	- Dave Fieldman	- Rolf Schuster

Thanks also must go to:

- ...the DOpus team, namely Dr. Greg Perry
- ...the StormC/Wizard team, namely Jürgen Haage and Markus Nerdling
- ...the Cloanto team

- ...the people on the DTA (Datatypes) Mailing List on Amiga World and the people of the "Datatypes Replacement Project" in general

This datatype also has been quite often quite high in the Aminet charts, so there hopefully soon will be many registered users... :-)

1.12 prefs

akPNGPrefs

akPNGPrefs is the Preferences Program for akPNG.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").

Icon by Bert Bosma <lmb@wx.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

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

OPTIONAL

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

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

"Datatypes/akPNG.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/akPNG.prefs_Tasks
Copy ENV:Datatypes/akPNG.prefs ENV:Datatypes/akPNG.prefs_Tasks/AWebIP
```

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

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

(Currently you can't edit them locally, sorry - but for e.g. AWeb it should be simple to manage that via ARexx by just including an appropriate script to AWeb's ARexx menu).

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) PROGRESSBAR=(ON|OFF)
- 7) SPEEDUP
- 8) CUSTOM_MODES
- 9) NOPPC
- 10) NOASPECT

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
PROGRESSBAR=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 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

Note: There's no need for BMF_DISPLAYABLE, don't rely on it.
And: If you encounter 'out of memory' or 'cannot open screen' problems, first try disabling interleaved bitmaps.

6) PROGRESSBAR

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

7) SPEEDUP (hidden option)

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

8) 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.

9) NOPPC (hidden option)

When the keyword NOPPC 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.

10) 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.

1.13 history

There's a Datatype Developers Mailing List (DTA) running now.
If you are a datatypes programmer and want to join, please

don't contact me for further information, but maybe ask
 info@amigaworld.com (Bodo) or simply visit the Datatype Association's
 WWW homepage under <http://www.amigaworld.com/support/dta/>

- 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>
 - There did occur problems with V36.126 of wizard.library, so you might wish to upgrade to V37.127 or higher (see Aminet:gfx/show/SViewNGWiz.lha). There's also a V39.101 available now - ask Haage and Partner. It's only used for the prefs program, so no need to worry, if you don't use the wizard version.
 - 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, too (1024). And make sure, you're using the latest FFS file system 43.x from www.amiga.de (it won't expire) - note, that you may update the FFS without repartitioning, but you have to be very careful when doing this from our HDTToolBox.

People, who did not receive their keyfile within 2-4 weeks after sending their registration should contact me.

History

=====

- V43.187 (5.6.98):
- updated akDT_Installer by Robert C. Reiswig
 (-> Robert C. Reiswig, akDatatype@vgr.com)
 - please consider to register...
 my gfxcard just broke :/
 - you have to handle a large number of images ?
 Organize, view, convert and process these ?
 Do you want speed up through PPC support ?
 Check out PMPPro, the universal graphics tool and picture manager. ↔
 - did you already notice, that SViewNG now has PPC native loaders/savers for e.g. PNG, JFIF, TIFF, PICT, too ?
 They're even faster than the datatypes, since they don't have to deal with OS bitmaps.
 - the Datatype Association (DTA) now has its own homepage (courtesy by Roger Hagensen)
-

and Amiga World) - it easily can be reached
under <http://www.amigaworld.com/support/dta/>

- V43.185 (7.5.98):
- if x/y aspect generation produces buggy results, e.g. with PictIcon, there's now an optional (and hidden) prefs switch to be set: "NOASPECT" will always let the datatype return 1:1 (-> Gunnar Schuster)
 - added internal replacement for BestModeID(); maybe it's still not perfect, but seems to work better than the original one, at least under CyberGraphX
 - fixed a bug in the HAM viewmode ID selection (related)
 - 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, too (1024). And make sure, you're using the latest FFS file system 43.x from www.amiga.de (it won't expire) - note, that you may update the FFS without repartitioning, but you have to be very careful when doing this fromout HDToolBox.
 - you have to handle a large number of images ? Organize, view, convert and process these ? Do you want speed up through PPC support ? Check out PMPro, the universal graphics tool and picture manager. ←
 - did you already notice, that SViewNG now has PPC native loaders/savers for e.g. PNG, JFIF, TIFF, PICT, too ? They're even faster than the datatypes, since they don't have to deal with OS bitmaps.
 - the Datatype Association (DTA) now has its own homepage (courtesy by Roger Hagensen and Amiga World) - it easily can be reached under <http://www.amigaworld.com/support/dta/>
- V43.181 (23.4.98):
- prefs program redone with StormC 2.00.32
 - prefs program now internally does include a copy of the .wizard file, so even if it isn't in PROGDIR:, the GUI can be opened (it just won't remember size and position, then)
 - added missing icons (-> CPurnell)

V43.180 (29.3.98): - PPC: - using libmoto.a now (faster!)

- upgraded to zlib 1.1.2
 - upgraded to libpng 1.0.1
 - updated/fixed prefs description
 - updated statement on DTM_WRITE support
 - updated docs
 - added new, improved Installer script by Robert C. Reiswig (-> a lot of credits hereby go to his address :-)
 - should be faster, have less bugs, run even more stable
- V43.175 (17.3.98):
- unified installation procedure, removed "Install" and "FIRST_Unpatch" scripts, added new Installer script by Robert C. Reiswig which now again does handle everything that's necessary for a proper installation
- V43.170 (9.3.98):
- PPC: - speed increase: using larger I/O buffers
 - improved code generation
 - forgot to bump revision in 43.165, now 43.170
 - 68k: - improved code generation
 - upgraded to zlib 1.1.1
 - upgraded to libpng 1.00 (the one before 1.0.0 ;)
 - now checks ENV:Classes/Datatypes/ for preferences, too
 - version history clean-up
 - added "Small PPC FAQ" to the FAQ section; it explains how to interpret speed-test results on different PPC systems
- V43.165 (7.3.98):
- 030/040/060 versions now generated from patch files
 - upgraded to zlib 1.1.0
 - upgraded to libpng 0.99d
 - added fixed "Install" script (thanks, Roger) (-> various)
 - PPC: - code should have become slightly faster
 - get ppc.library 45.20 from ftp.phase5.de ...
 - fix of the "custom mode" mechanism *may* follow in an upcoming release; no need to continuously report bugs!
- V43.160 (23.2.98):
- PPC: - fixed memory management
 - added version string
 - increased stacksize to 16K (fromout 68k side)
 - get ppc.library 45.17 from ftp.phase5.de ...
 - 68k: - adjusted to new PPC version
 - fixed possible small bugs (calling PPC)
-

- added hidden prefs option "NOPPC" to allow disabling of the PPC decoder (e.g. for specific tasks). Useful e.g. for speed comparisons.
 - updated comments on (Fast)IPrefs, added note about FastIPrefs/PPC problems (MCP-related ?)
- V43.155 (13.2.98):
- removed PPC support in 68030 version (040/060 only now)
 - PPC module now works with ANY program, not only AWeb
 - you still need a keyfile to make full use of the PPC decoder, but without a keyfile it now basically does work, too: it then switches to a DEMO MODE, where the whole image is loaded and decoded, but only each 3rd line actually exported (so you can guess how fast it is, but you won't be able to take full advantage from it without registering)
- V43.150 (4.2.98):
- cleaned up source code
 - better modularization
 - reduced number of file access during handling of preferences settings (now just opened and read once)
 - various optimizations and fixes to the source code
 - added hidden prefs option "SPEEDUP", which just activates the bitmap related optimizations done in V43.135; it's no longer default, since users running a DraCo may have encountered problems with picture.datatype V42 (V43 BTW suffers from different problems)
 - added PPC support via an ELF plugin module for PowerUP (TM): it will only work with AWeb and only for registered users, though. If you already are a registered user, you can receive the necessary keyfile via email on request. Note, that I won't send any keyfiles via snail mail in the future, so always mention your email address! See FAQ.
(You may have to install the PPC module by hand, BTW.)
 - decoder now single-threaded due to some strange problems when under heavy parallel use
(-> Troels Walsted Hansen)
 - upgraded to zlib 1.0.8
 - upgraded to libpng 0.99
 - fixed bug in colormap initialization for <= 8 Bit (should not have had any impact actually, though)
- V43.140 (25.1.98):
- upgraded to libpng 0.97
 - ...and libpng 0.98, while Aminet was down
 - additionally just upgraded to zlib 1.0.7
 - full recompilation
 - added new icon for prefs program (NewIcons)
(-> by Bert Bosma <lmb@wxs.nl>)
 - fixed enforcer hit as introduced in 43.135
(-> Sebastian Becker, Thomas Tavoly, various)
 - added some notes about transparency to the "FAQ"
 - tried some more fixes on the "custommode" code parts
(-> Luca Longone)
 - we looked for "CUSTOM_MODES" in akJFIF.prefs
-

- instead of akPNG.prefs
 - you now can describe a custom mode as text OR as a hexadecimal value
- V43.135 (5.1.98):
- optimized progress bar code
 - fixed possible bugs in the progress bar code
 - upgraded outdated "1997" to "1998" bugfix release ;-)
 - made bitmap allocation and handling a little bit more cgfx-aware
 - added special speed hack for 8 Bit output on AWeb-II screens under CGfx
- V43.130 (12.12.97):
- speed: optimized compiler's code generation
 - "beta" text was left in ID string of V43.121
- V43.121 (23.11.97):
- OLD_DTM_WRITE switch was taken from akJFIF.prefs :-(
 - the new dispatch routine again (or still) has been causing problems, this time with IBrowse (not displaying certain images, maybe because of refresh problems). Removed all the new code, again using David N. Junods good old method, here (-> Philippe Duchenne)
- V43.120 (11.11.97):
- Prefs-GUI now with correct version id
 - included fixed "install" script by Roger Hagensen, which - besides some other bug fixes - now also does check for the GUI version (-> several)
- V43.110 (15.10.97):
- 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'
 - Prefs program accidentally linked with storm020.lib. Fixed ↵.
- V43.100 (5.10.97) :
- rewrote Dispatch() routine completely, implemented new routines derived from Roland Mainz' improved sample code
 - DTM_WRITE now can be forced to its old behaviour by specifying the hidden prefs option "OLD_DTM_WRITE" (IFF-ILBM then will be written without any error code given because of unsupported DTWR_RAW)
 - recompiled prefs program with StormC 3.0 (2.00.23)
 - prefs now using external .wizard file (improve GUI, if you like)
 - prefs GUI now remembers position and size
- V43.97 (19.09.97) :
- transparency handling was broken
 - updated docs
- V43.96 (02.09.97) :
- seems as if "CUSTOM_MODES" had been broken :-/
-

- updated docs
 - V43.95 (12.08.97) : - recompiled with SAS/C V6.58
 - V43.92 (28.07.97) : - updated docs
 - updated 680x0 info
 - updated akDT_Installer by Robert C. Reiswig
(-> Robert C. Reiswig, akDatatype@vgr.com)
 - recompiled prefs program with StormC V3.0
 - fixed possible, `_small_` memory leak within progress window's msg handling
 - V43.91 (04.07.97) : - fixed "memory loss on LibInit failure" bug
 - added new akT V43.70
(-> Roger Hâgensen)
 - since after the CUSTOM_MODE addition, the bestmode routine looked quite ugly and was somewhat redundant. Fixed.
 - updated scripts
(-> done by Roger Hagensen)
 - progress bar: replaced two Forbid/Permit by semaphore
 - removed another (unnecessary) Forbid/Permit pair
 - some `__saves` were missing
 - `__inlined CalcProgressPos()`
(not time-critical)
 - V43.90 (24.06.97) : - V43.81 should had been available as binary-only fix archive "akFix4381.lha" together with the other datatypes, only.
Upload did not succeed
(-> a zillion people did report V43.80 bug)
 - added akDT_Installer by Robert C. Reiswig
(-> Robert C. Reiswig, akDatatype@vgr.com)
 - added note about mathffp.library patch program 'ffppatch' (Aminet:util/boot/)
 - V43.81 (16.06.97) : - new dispatcher code either still is buggy, only works with V45 or has been implemented wrongly. Anyway: V43.80 was messed up, better use V43.81 !!
 - V43.80 (14.06.97) : - SetPatch 43.6 is on ftp.amiga.de
Get it!
 - CUSTOM_MODES now (optionally) accepts a fixed hex value as viewmode settings
(-> Joel Alvim)
 - protection bits now correctly set
(-> Per Jonsson)
 - updated class dispatcher
(-> Roland Mainz)
 - V43.75 (27.05.97) : - HAM indicator seems to have been broken since some releases (viewmode field, HAM_KEY)
 - added new viewmode generation routine, especially for HAM modes, which takes care of AGA HAM
-

- capabilities and PAL/NTSC
 - aspect ratio fields of BitMapHeader structure now are filled according the selected viewmode
 - added new akT V43.70
(-> Roger Hågensen)
- V43.70 (17.05.97) : - upgraded to libpng 1.0 beta 6 - version 0.96, which has been said to fix some serious bugs like:
- bug with <8bpp images introduced in 0.95
 - 256-color transparency bug
 - etc.
- detected a makefile bug, which might have caused only the 68000 version to have been upgraded to V0.95 in 43.60. The others perhaps were still V0.90 based ?!
- added hidden prefs option, which generates all viewmodes out of the standard set (-> Joel Alvim):
- LowRes (320x200/256)
 - HighRes (640x200/256)
 - SuperHighRes (1280x200/256)
 - LowRes Lace (320x400/512)
 - HighRes Lace (640x400/512)
 - SuperHighRes Lace (1280x400/512)
- (-> Joel Alvim)
- V43.61 (04.05.97) : - minor changes
- V43.60 (20.04.97) : - upgraded to libpng 1.0 beta 5 - version 0.95
- updated email list
 - updated docs
- V43.50 (30.03.97) : - in the prefs, "24 Bit" was "Dithering" and vice versa (V43_MODE)
- (-> Milco Veljanoski)
- V43.40 (13.03.97) : - due to problems with the (global and custom) startup the prefs program did only run from CLI/Shell, not WB. Could not explain that at first, fixed now.
- (-> Tom Lively, Torbjörn Aronsson, Max Headroom, Ian Barclay, Marc-Tell Volkmann, Christian Beck)
- note, that a copy of wizard.library also can be obtained from: Aminet:gfx/show/SViewNGWiz.lha (library only)
 - redone with StormWizard 2.0
 - etc.
- V43.30 (03.03.97) : - now does recognize bit_depth / num_palette differences and consequently reduces color depth for 2..256 color pictures, when necessary (saves some more (chip) memory and bandwidth, already should have been in last release)
- now, finally the GUI of the prefs program has been re-designed with StormWizard, so this program needs "wizard.library" V37+ (you can find a copy in the StormC Demo archive or the Wizard Demo package on Aminet under "biz/demo/StormWIZARD1_0.lha").
- Note, that the GUI still is not very sophisticated,
-

- but at least sizeable and font-sensitive now.
- prefs program now developed/compiled with StormC 2.x

- V43.27 (23.02.97) : - MAJOR REVISION
- there's now a global BestModeID() patch available as Aminet:util/sys/ModeP.lha
 - there were problems with 16 Bit grayscale images (-> Magnus Holmgren)
 - the 16 Bit bug was a side effect of a wrongly handled pixel_depth check, thus perhaps it would have happened with other bit_depth 16 images as well
 - removed some misinterpretable information on 16 bit (highcolor) modes, which are not allowed with PNG, with the exception of 16 Bit grayscale images (-> Magnus Holmgren)
 - added an overview table of valid PNG depth combinations (referring to the specification)
 - re-introduced "wm32 magic fix"
 - __inline'd HAM code also, now
 - there were "memory losses" under specific conditions (below is what I found out) (-> Magnus Holmgren)
 - under low memory conditions there might have appeared even more memory loss, since setjmp'ing of the PNG sources did cause some high level allocations not to be delocated - this mainly did concern the non-V43 modes
 - gfx mode selection now a little bit more AGA sensitive, ModeP patch still recommended, though
 - tested compatibility by viewing Willem van Schaik's "PNG Test Suite" in V40 ordered dithering 8 Bit mode (<http://www.wco.com/~png/pngsuite.html>). When there is enough (chip) memory, all pictures seem to be handled and displayed correctly (now). On the page there are samples for all types of images (depth combinations) as mentioned in the PNG Suite Documentation. If you find anything not displayed correctly, though: please tell me. Note, that gamma correction is not applied (no need to mention).
 - 2..128 color images no longer imported as 256 color images (2 still as 4, though), this also saves (chip) memory... (-> Roger Hågensen, various)
 - fixed bug in reading 24 Bit + alpha channel images in V40 HAM8 or ordered dither mode (HAM6 worked fine)
 - added some notes to the docs
 - most stable version so far; nearly rock-solid
- V43.26 (15.02.97) : - skipped V43.16-25
- fixed installer script's "copy 040 version" bug (-> Philippe Duchenne, Tom Lively)
 - added special note about interlaced files and progressive display, since some people
-

- still did not get it right
- superview.ftn.sub.org will be replaced by ftn.neckar-alb.de until 22.2.97
- there was a bug in interlaced reading, which caused isolated black pixels (at end of 8x8 matrix) with some graphics - this did not appear in V43 mode for some reason, and was not so obviously detectable in V40 HAM-mode (-> Vulture)
- temporary storage management should be somewhat faster now, since the top-level access functions now also have been `__inline'd` (only concerns interlaced graphics)
- rewrote docs
- removed charts

V43.15 (05.02.97) : - V43.14 did screw up. Recompiled.
- checking for PNG_INFO_tRNS to recognize transparency

V43.14 (04.02.97) : - now using libpng 1.0 beta 4 - version 0.90

V43.13 (30.01.97) : - there were not much registrations yet.
HAVE YOU CONSIDERED TO REGISTER ?
(spending lots of hours a week on this stuff)

- improved error handling
- now explicitly checks `DTA_SourceType == DTST_FILE`
- forgot to bump version
- a "Prefs/Datatypes.info" was missing (-> Roger Hâgensen)
- temporary file management was buggy (no crash, but incorrect output for some kinds of images) (-> Vulture)
- temporary storage management was extremely slow, because it did use files always and did not make use of buffered I/O:
 - temporary storage access has been capsuled into four intelligent functions, which select the buffer medium automatically (`T_Open`, `T_Close`, `T_Read`, `T_Write`)
 - now only switches to file mode, when there is not enough RAM available for buffering (basically like with akJFIF's VMEM: handling)
 - now uses buffered I/O when in "file" mode (adjusted to needs by appropriate `SetVBuf` call)
 - temporary storage would have aquired stack even if not active and unused (256 bytes)
 - uses less stack space when in "ram" mode (another 256 bytes, temporarily used)
 - should solve interlaced problems, as well (removed the corresponding "problems" section)
- added new akT version by Roger Hâgensen
- a `pr_WindowPtr` of -1 was not recognized as NULL ptr (-> Dr. Greg Perry)

V43.12 (25.01.97) : - reworked docs
- added new installer script and special "akT" version of AmigaT tool by Roger Hâgensen

- (thanks, Roger).
 - prefs program moved from sys:prefs to sys:prefs/datatypes
(-> Roger Hâgensen)
 - would always have exported 256 colors in V40 8 Bit mode
 - minimum bitmap depth set to 4 (compare akGIF)
 - various small changes to the code
 - now, in a first attempt, checks PDTA_Screen with highest pri for where to open the progress window - only works, when always remapping to the same screen (e.g. some web browsers)
(-> Mat Bettinson)
 - note: get and install SetPatch 43.4 or 43.5
- V43.11 (15.01.97) : - installer script was faulty
(-> Harry W. Turner II)
- added "http://193.203.162.219" info
(-> Harry W. Turner II)
 - tmp files would not have been deleted from VMEM:
 - guide now is interactive and connects URLs to AWeb-II if available
- V43.10 (09.01.97) : - added new installer script and "AmigaT" tool by Roger Hâgensen (thanks, Roger). Only slightly modified script.
- added script icon (using IconX), may nevertheless still be started from shell
(-> Roger Hâgensen)
 - added directory icon
(-> Roger Hâgensen)
 - added note about new homepage
 - completely rewrote docs
 - added "PNG := successor of GIF" note, and explained, that free algorithms are as free as free speech on the internet
- V43.9 (01.01.97) : - reworked docs
- added ordered dithering support for V40 mode (16..256 colors)
 - now allows to run it in V40 mode even with picture-datatype V43, thus HAM6/8 output becomes possible
 - (hopefully) fixed progressive reading, when in V40 HAM6/8 mode, also applied the same fix to ordered dithering: in this case a temporary file must be created at VMEM:
 - added support for task-specific settings (private settings for specific programs)
 - added "transparency" note. Please read.
 - now using zlib 1.0.4
 - 8 Bit only:
now by default uses interleaved bitmaps, which should fix the remaining problems with small sized bitmaps and/or distorted bitmap borders. Can be switched off.
-

- V43.8 (17.12.96) : - installer script now asks for CPU type and selectively installs the single datatype versions (-> first version worked out by Roger Hågensen)
- now using libpng 1.0 beta 3 - version 0.89
- now using zlib 1.0.3
- slightly reduced size (about 4 K each)
- better compiler optimization on pnglib (esp. 040/060)
- better compiler optimization on zlib (esp. 040/060)
- using new libpng API
- less PNG bugs (libpng)
- interlaced (progressive) 24 Bit graphics under pic-dt V40 would have caused distorted HAM6/8 display. Now just "bad".
- V43.7 (7.12.96) : - updated/changed docs
- did not create correct mode ID for HAM6 display
- fixed possible memory leak on temporary memory allocation error
- V43.6 (30.11.96) : - updated/changed docs
- fixed small bug (GadTools)
- fixed CTRL-C problem (IBrowse)
(-> Matteo Tenca)
- V43.5 (24.11.96) : - progress bar now treated like requester. Now will check pr_WindowPtr before trying to open on (default) pubscreen
(-> Edwin H. Bielawski)
- added special fix for AWeb 2.1, which has pr_WindowPtr correctly set, but does let the Datatype operations do by an "AWebIP" task, which has not. Thus when being called by "AWebIP" we now refer to the pub screen "AWeb" instead of fiddling with pr_WindowPtr
- V43.4 (23.11.96) : - progress bar window no longer auto activated
(-> André Laemmer)
- when called by IPrefs (e.g. for WBPatters), a progress bar will never pop up
- did not correctly read interlaced (progressive) PNGs
(-> Andreas Mixich)
- on missing picture.datatype V39, prefs program would have claimed about missing V40
(-> Robert Wahnsiedler)
- now exports transparency information (256 color graphics only)
- V43.3 (12.11.96) : - completely recompiled with SAS/C V6.57
- added 68030, 68040 and 68060 version
(-> Ingo Jürgensmann)
- there was a broken version of V43.2 (3.11.96) (in parts still labeled as V43.1) which unfortunately went on Aminet, but hopefully had been
-

- overwritten by the right version later on
- added doc section about 680x0 CPUs and FPU's. Please follow the instructions given and use the FMath patches, also SetPatch 43. The latest version floating around was SetPatch V43.5 - which is currently not on ftp.amiga.de, but maybe available elsewhere
- progress bar did display "akJFIF" instead of "akPNG"
- changed PubScreen behaviour of progress bar (-> Thomas Boerkel)

V43.2 (3.11.96) : - bugfixes

V43.1 (1.11.96) : - first release (with V43 support)

1.14 About PNG - successor of GIF

PNG is the successor of the GIF file format. 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.

1.15 PNG-Box - WWW tool for PNG writing

PNG-Box

- SHAREWARE -

© 1997-98 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 conform and based on wizard.library.

See program archive for copyright and distribution information.

See Aminet:gfx/conv/PNG-Box.LHA for download.
