akPNG_Documentation

COLLABORATORS

	TITLE : akPNG_Documentation		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		July 8, 2025	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

akPNG_Documentation

1.1 akPNG : Documentation

```
                         akPNG.datatype V44.6

                             - SHAREWARE -

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

                    A PerSuaSiVe SoftWorX PRODUCT.

                        Needs Kickstart V3.x

                     Release Date : 16.9.1998

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


           <Commercial> BTW: What is SViewII ? </Commercial>
           <Commercial> Already tested PMPro ? </Commercial>
```

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

```
                              Please visit:

                             WWW Support Site
                        http://wdo.de/ark/ (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

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 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.

1.5 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 46070090 Kto. 0298174

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 akPNGPrefs 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 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 PNG datatype, which is based on the
 latest PNG sources (zlib V1.1.3, libpng 1.0.2).

 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.
```

1.7 Datatype FAQ

```
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 (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, 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 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.

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 explicitely 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 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

  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 "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 speficic 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
  transpareny 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 correcty.
  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.

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...

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

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" (V8 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.

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' mathieee-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 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.
```

1.9 correspondence

```
** General PerSuaSiVe SoftWorX WWW Support Site is http://wdo.de/ark/
** - actually redirected to http://home.t-online.de/home/Andreas_Kleinert/

  __________________________________________________________
 |      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.          |
 |__________________________________________________________|

EMail:

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

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

         - Fido   Andreas Kleinert 2:2457/350.18
         - Usenet
            >>>   Andreas_Kleinert@t-online.de     (T-Online)
                  ARK@News.wwbnet.de               (Z-Netz)
                  ARK@superview.ftn.neckar-alb.de  (Fido-Gate)
                  Andreas_Kleinert@gmx.de          (GMX)

           (note, that mail sent to @gmx.de currently will be
            forwarded to @t-online.de - so, as long as it works,
            try to address the latter directly)

         - If nothing else works, try one of these public
           Fido-Usenet gateways:

             In Germany:
               Andreas_Kleinert@p18.f350.n2457.z2.fido.sub.org

             From USA or elsewhere:
               Andreas_Kleinert@p18.f350.n2457.z2.fidonet.org
```

1.10 thanks

```
Thanks go to (in order of appearance ;-)
========================================

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

```
- Ingo Jürgensmann     - Thomas Boerkel        - Andreas Mixich
- Robert Wahnsiedler   - André Laemmer         - Edwin H. Bielawski
- Matteo Tenca         - Jan Skypala           - Adrian Demarais
- Ludwig Berndt        - Roger Hâgensen        - 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 Rottländer
- 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. Küsche
- Jörg Handwerg        - Stefan Michel         - Jochen Rhein
- David Newman         - Bradley Rogers        - Simo Koivukoski
- Michael Jaccoud      - Jan Uerpmann          - Achim Akkermann
- David Gill           - Willi Demuth          - Sander Assenbroek Machielsen
- 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
- Andrew Zalotocky     - Mark Carter           - Thomas Steffens
- Carsten Knodel       - Emmanuel Rey          - Sven Ottemann
- Matthias Laskowski   - Ralph Ewers           - Thomas Wiedecke
- John Jackson         - Robin Hüskes          - Vincent Morenas
- Neil Bothwick        - Javier Marcet         - Michael Merkel
- Ralph Ewers          - Steve Krueger         - Jim Cooper
- Clifford Mould       - Jon Steinar Kvaale    - Jon B. Peterson
- Oliver Molz          - Klaus Müller          - John Aadnoy
- Sven Bornkessel      - Arvid Schlesinger     - Armin Klippel
- Wolfgang Krause      - F. Ruthe              - Alexander Niven-Jenkins
- Gary Goldberg        - Thomas Birk           - Vincenzo Morra
- Holger Kruse         - Michael Burkhardt     - Keith Blakemore-Noble
- Alan Surrette        - 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


Thanks also must go to:

  - ...the Amitrix team, namely Brant Coghlan and Dale Currie
  - ...the Cloanto team, namely Michael C. Battilana
  - ...the DOpus team, namely Dr. Greg Perry and Jonathan Potter
  - ...the Haage and Partner people, namely Jürgen Haage and Markus Nerding
  - ...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

```
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@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


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.

  As with V44.1 this no longer needs to be done by hand, but easily
  can be managed fromout the (original) preferences program (as long
  as the corresponding task actually is running at the same time).
  ---------------------------------------------------------------------

  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) NOASCPECT

  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. EGSPlus

  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.
```

1.12 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>

             - There did occur problems with V36.126 of wizard.library,
               so you may wish to upgrade to V37.127 or higher (see
               Aminet:gfx/misc/SvII-WIZ.lha). There also are newer
               versions available, but obviously not on Aminet - ask
               Haage and Partner or check their latest demo version
               releases. -- Since it's only used for the prefs program,
               there's no need to worry, if you don't use the wizard
               version, though...

             - viewmode selection may not alway 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 fromout HDToolBox.


 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
 =======
 V44.6 (16.9.98):   - 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. Please have a look!
                    - added modified Prefs GUI by Georg Rottlaender
                    - the newest wizard.library version seems to be V40.101;
                      you can find 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...
                    - PPC mode: fixed memory leak in "ordered dithering" mode,  ↩
                       which
                      could have taken place when memory was very low;
                      did not happen during normal usage
                    - there's now another 68k/PPC datatype available: akTIFF

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 PNGs with upto 256 colors
                           now will be moved faster to their
                           destination bitmaps (registered version,
                           only). Does not apply to dithered 24 bit.
                    - removed history entries for versions below 44.1
                    - hey, did you ever imagine do own a LEGAL keyfile
                      for this software product ? Register NOW and
                      take a look into the CHEAP Discount offers, please!

V44.4 (9.8.98):     - *** MAJOR RELEASE ***

                      This version seems to be quite stable now.
                      Updates will appear when necessary.

                    - checked using the PNG Suite test icons
                      and found some bugs
                    - 68k: ordered dithering of _interlaced_
                      true color images with alpha channel
                      resulted in black images
                    - long-standing bug: gray scaled images
                      (in 8 or 16 bit) _with_ 8 or 16 bit
                      alpha channel would not have been
                      read correctly; some strange kind of
                      colored true color image would have
                      been exported. On the PPC side, even
                      crashes (through damaged memory list)
                      were possible. Perhaps the same problems
                      could have happened with 8 bit colormapped
                      files plus alpha channel, not sure.
                    - in normal 8 bit mode, a temporary bitmap
                      wasn't released. Caused a small memory loss.
                      (-> Troels Walsteds Hansen)
                    - fixed small problem in PPC startup code
                    - 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 explicitely clearing those lines,
                      which MIGHT slow down the PPC demo mode
                      when compared to the registered PPC mode.
                    - PPC: while fixing this, noticed, that
                      the 32 bit (24 bit + alpha channel) mode
                      never would have worked as expected.
                      Nothing was written to the V43 bitmap.
                      Additionally, there was another, more
                      general problem with alpha channels in
                      PPC mode - causing bad output in all
                      modes. The same goes for *interlaced*
                      alpha channel graphics (another bug).
                      (yes, really - three bugs in one routine;
                      2 on the 68k side, 1 on the PPC side)
                    - 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):    - upgraded to zlib 1.1.3
                    - added info on new, permanent Shareware discount
                      (bundling) offer
                    - removed commercial header in guide file
                    - 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
                    - iffparse.library was opened although not needed
                    - 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):         - upgraded to libpng 1.0.2
                    - completely recompiled PPC part and 68k png parts
                    - 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. 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 -

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

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