

akPNG_Dokumentation

COLLABORATORS

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

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	akPNG_Dokumentation	1
1.1	akPNG Dokumentation	1
1.2	rechtliches	2
1.3	haftung	3
1.4	vertrieb	4
1.5	registrierung	4
1.6	Installation und sonstige Hinweise	4
1.7	Benutzung von 680x0 CPUs und PPC-Turbokarten	7
1.8	kontakt	9
1.9	danksagungen	10
1.10	Datatypes FAQ	13
1.11	Über PNG - den Nachfolger zu GIF	22
1.12	PNG-Box - Ein InterNet-Tool zum Schreiben von PNG	23
1.13	versionsgeschichte	23
1.14	bedienung	25

Chapter 1

akPNG_Dokumentation

1.1 akPNG Dokumentation

akPNG.datatype v44.112

Veröffentlichungsdatum: 14.08.2001

Benötigt Kickstart V3.x

© 2001 by Andreas Ralph Kleinert.
Alle Rechte vorbehalten.

EIN PerSuaSiVe SoftWorX PRODUKT.

- SHAREWARE -

Bitte lassen Sie sich registrieren - weniger als 1% der Benutzer
eines Programmes lassen sich Registrieren. Das ist nicht viel.

<Commercial> Was ist SViewIV ? </Commercial>

Rechtliches
Haftung
Vertrieb

Registrierung

Installation
Bedienung

Free algorithms... PNG - Nachfolger von GIF ...and free speech !

Datatypes FAQ
68020-68060, PPC

Kontakt

Danksagungen
Versionsgeschichte

Only $\overline{\backslash X} / \overline{\text{Amiga}}$ makes it possible.

Die CHAOS Theorie:

"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)

Ähem... nun ja:

... und Danke für den Fisch.

1.2 rechtliches

Das Programm in dieser Version und auch seine Dokumentationsdateien
unterliegen dem (C)opyright 1996-2001 von Andreas Ralph Kleinert.
Alle Rechte vorbehalten.

Das Recht zur Nutzung des Programmes wird durch Zahlung einer
SHAREWARE-Registrierungsgebühr an den Autor gewährt.

Diese Software basiert in Teilen auf der PNG Reference Library (inkl.
libpng und zlib), welche eine freie und auch kommerzielle Nutzung zulässt.

libpng:

libpng 1.0.12
Copyright (c) 1995 - 1996 Guy Eric Schalnat, Group 42, Inc.
Copyright (c) 1996 - 1997 Andreas Dilger
Copyright (c) 1998 - 2001 Glenn Randers-Pehrson

zlib:

zlib 1.1.3
(C) 1995-1998 Jean-loup Gailly and Mark Adler

Der akDT_Installer ist (C)opyright 1996-2001 von Robert C. Reiswig.
Falls Sie irgendeinen Teil dieses Installationsscriptes benutzen möchten, fragen
Sie ihn. Ohne Genehmigung darf es nicht in eine andere Distribution eingebunden,
oder dort genutzt werden! Änderungswünsche, Vorschläge oder Probleme bitte an
folgende EMail-Adresse senden: akDatatype@vgr.com

Der WarpUP Elfloader (ElfLoadWOS) Quelltext ist im Original von

Peter Annuss <paladin@cs.tu-berlin.de>

Dieser wird für das Laden und Ausführen der EGCS 2.91.57 WOS PPC Programme unter AmigaOS benötigt. Er wurde komplett neu geschrieben und enthält jetzt sogar einige Erweiterungen und Fehlerbereinigungen.

Siehe auch <http://cs.tu-berlin.de/~paladin/> für weitere Informationen.

Das Oberflächendesign des Voreinstellungsprogrammes wurde von

Georg Rottlaender <Georg.Rottlaender@bonn.netsurf.de>

unter Benutzung von 'NewIcons' Grafiken von

Philip Vedovatti <vedovatt@u.washington.edu>

verbessert - mit Erlaubnis des 'Team NewIcons'

Die Patch-Dateien wurde unter Benutzung des Programmes scompare SAS Binary File ←
erstellt.

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.

Einige der genannten Namen oder Produkte innerhalb diesem oder anderer Dokumente unterliegen dem Copyright der Hersteller oder der jeweiligen Person.

Sollten einige der aufgezählten Bedingungen und Klauseln innerhalb dieses Dokumentes in Verbindung mit den Rechtssprechungen einiger Länder nicht gültig ←
sein,
so führt dies nicht zur Ungültigkeit der anderen Klauseln.

1.3 haftung

Der Autor übernimmt keine Verantwortung für irgendeine Art von Auswirkungen, die durch die Nutzung dieses Programmes oder eines Teiles davon entstehen.

Diese Software wird zur Verfügung gestellt "WIE SIE IST" und es wird keine Garantie irgendeiner Art übernommen, so daß Sie diese Software auf eigenes Risiko nutzen.

Der Autor behält sich das Recht vor, dieses Programm nicht weiterzuentwickeln.

1.4 vertrieb

Diese Version des Programms kann frei vertrieben werden (SHAREWARE). Sie können das Archiv vervielfältigen, solange die Copyright Dokumente nicht geändert werden und alle anderen Teile dieser Distribution enthalten sind.

Dieses Programm darf nur mit meiner schriftlichen Erlaubniss in kommerziellen Archiven oder kommerziellen Programmen genutzt werden – Fragen Sie einfach danach!

Dieses Programm darf auf Public Domain Disketten oder in Public Domain Bibliotheken enthalten sein. Für die Verbreitung mittels diesem Weg, darf eine kleine Spende verlangt werden, welche die Kosten für die Erstellung der Kopien decken kann. Es ist aber auf die oben beschriebene Einschränkung der kommerziellen Nutzung zu achten.

Dieses Programm darf auch über EMail vertrieben werden und in Mailboxen enthalten sein, solange die Weitergabebestimmungen in allen Punkten eingehalten werden.

Bei der Benutzung oder Weitergabe dieses Programmes erklären Sie sich automatisch mit allen oben aufgeführten Konditionen und Bestimmungen einverstanden.

1.5 registrierung

Sie können mir das Geld in einem Umschlag zusenden, EC-Schecks nutzen, oder einfach 15 DEM (etwa 10 US\$) Shareware Gebühr auf das folgende Konto überweisen (bitte vermerken Sie Ihren Namen!):

Deutsche Bank Siegen	
BLZ	460 700 24
Konto	0298174
SWIFT	DEUTDEDK460

Bitte keine Fremdwährungsschecks (EC-Schecks oder DM-Schecks sind ok).

1.6 Installation und sonstige Hinweise

GIF ist veraltet - Sie sollten es zukünftig weder nutzen noch unterstützen. Wenn Sie WWW-Design machen, nutzen Sie stattdessen PNG und JPEG. Es ist wichtig!

Installation und Benutzung

Installieren Sie einfach die Datatype Dateien in ihre jeweiligen Verzeichnisse und kopieren Sie das Voreinstellungsprogramm alternativ nach

SYS:Prefs/Datatypes

Dadurch, daß das Datatype selbst irgendwo plaziert werden kann, solange dorthin ein gültiger Suchpfad existiert, MUSS das PPC-Modul nach SYS:Classes/Datatypes/ kopiert werden! Dies ist kein Problem, solange Sie das Installationsscript benutzen, ansonsten denken Sie bitte daran...

Bitte stellen Sie außerdem sicher, daß ein Verzeichnis zur Ablage temporärer Dateien verfügbar ist. Es muß hierauf ein Assign namens "VMEM:" existieren. Wie übrigens auch bei der superview.library und dem akJFIF.datatype .

Erstellen Sie kein Assign auf "T:", sollte dieses auf die RAM-Disk verweisen (dies ist auch der Grund, warum das nicht schon standardmäßig geschieht) - nehmen Sie einfach einen sicheren Platz dafür (VMEM: ist gemeint ;).

Programm Information

Das akPNG.datatype ist ein PNG-Datatype, welches auf den aktuellsten PNG Quellcodes basiert (zlib V1.1.3 und libpng V1.0.2).

Mit V39-42 des picture.datatype wird eine (bis zu) 256 Farben große Palette erzeugt (entweder Ohne, mit Floyd-Steinberg oder mit Geordneter Rasterung), oder HAM6/8 Ausgabe.

Mit dem picture.datatype V43 werden auch 24bit unmodifiziert exportiert.

Vom picture.datatype V43 sind Versionen für CyberGraphX und Picasso96 verfügbar, wobei die Picasso96 Version auch ECS/AGA unterstützt - benutzen Sie einfach die für Sie sinnvollere von beiden.

akPNG benutzt bei Verfügbarkeit Memory Pools und automatisch auch die asyncio.library (V39+), sollte diese verfügbar ist.

Für die besten Einstellungsmöglichkeiten wird empfohlen, das beigefügte Voreinstellungsprogramm zu benutzen. Natürlich kann auch ein alternatives Programm aus dem Aminet benutzt werden, welches die gleiche Funktionalität beinhalten sollte (aber bitte denken Sie daran mir dann keine Fehlerreports zu senden!). ↩

Das akPNG.Datatype ist SHAREWARE, die Zukunft liegt bei IHNEN.

Bekannte Fehler

Einige Leute berichteten in der Vergangenheit über Probleme mit dem Installationsscript. Wenn Sie einen Fehler feststellen können oder Probleme damit haben, so senden Sie eine Nachricht direkt an den Autor des Scriptes:

Robert C. Reiswig <akDatatype@vgr.com>

Bitte benutzen Sie mindestens die wizard.library V41.101. Sie sollten eine Kopie davon in den verschiedenen Demoversionen von Programmpaketen des Herstellers Haage&Partner finden können:

[ftp.haage-partner.com](ftp://ftp.haage-partner.com)

Die Auswahl des Anzeigemodus ist nicht immer 'perfekt'.

Tipps und Tricks

Wenn Sie dieses Datatype in Ihrem WWW Browser benutzen sollten, dann erzeugen Sie eine separate Partition (Größe etwa 30-70 MB) um temporäre Daten speichern zu können und verweisen (Assign !) Sie mittels VMEM: dorthin. Den Cache Ihres Browser's verweisen Sie bitte auch dorthin.

Versichern Sie sich zusätzlich das Addbuffers größer/gleich 128 ist. Beim Partitionieren (ACHTUNG: Sie verlieren dabei die dort stehenden Daten!) macht es Sinn, die Blockgröße auf einen höheren Wert als vorgegeben zu stellen, z.B. 1024.

Stellen Sie außerdem sicher, daß Sie die neueste Version des FFS Filesystems von www.amiga.de benutzen!

Bitte beachten Sie, daß Sie das FFS ohne eine Neupartitionierung updaten können. Aber Sie sollten sehr aufpassen, wenn Sie dies mit dem Programm HDToolBox machen.

Noch besser ist es, ein schnelleres Filesystem für Ihre Cache-Partition zu benutzen, wie z.B. das kommerzielle PFS2 (ehemals AFS, www.schatztruhe.de), oder das frei erhältliche SFS (siehe <http://www.xs4all.nl/~hjohn/SFS/>).

Keyfile Probleme:

Leute welche nicht innerhalb von 2-4 Wochen ihr Keyfile nach versandt ihrer Registration erhalten, sollten mich kontaktieren (während des Sommers beachten Sie bitte, daß es keinen Sinn macht sich schon nach 2 Wochen zu melden - einige Menschen tendieren dazu manchmal Urlaub zu machen...)

1.7 Benutzung von 680x0 CPUs und PPC-Turbokarten

Grundsätzlich läuft dieses Programm schon auf einer einfachen 68000 CPU.

Wie auch immer, wenn sie eine 68020/030+68881/882 FPU oder eine 68040/060+FPU, oder vielleicht sogar eine PPC-Karte mit zwei Prozessoren haben, wünschen Sie sich ganz sicher auch die extra Pferdestärken nutzen zu können.

Hierfür gibt es ein paar Einstellungsmöglichkeiten, spezielle Libraries und/oder sogar Patches. Sie sollten sich vielleicht in diese Sache ein wenig tiefer einarbeiten – aber seien Sie vorsichtig dabei!

PPC Unterstützung

=====

1. Bei CyberStormPPC Karten macht es Sinn die Tools "SetFastAvec" und "Set60nsMode" (SetMemMode) einzusetzen, welche die Leistungsfähigkeit Ihres Systems erhöhen können, z.B durch das Ansprechen des RAM mit 60ns anstatt von 70ns.

Neuere Version erlauben das Ansprechen dieser Einstellungsmöglichkeiten schon vom Bootmenü der Karte aus. Sollten Sie dabei zufällige Abstürze erhalten, dann schalten Sie einfach wieder zurück in den 70ns Modus.

2. Versichern Sie sich, daß Ihre Turbokarte über genügend RAM verfügt, sodaß der PPC nicht gezwungen ist das weitaus langsamere RAM auf der Hauptplatine anzusprechen. Sollten Sie zufällige Abstürze erhalten, dann versichern Sie sich, daß Sie den Installationsanweisungen genau gefolgt sind und nicht verschiedene SIMMs von verschiedenen Herstellern als eine 64bit Speicherbank konfiguriert haben.
3. Dieses Programm benutzt die "ppc.library". Deshalb versichern Sie sich, daß Sie

- a) die "powerpc.library" NICHT installiert haben oder
- b) eine Version der "powerpc.library" installiert haben, welche keine Probleme in Zusammenarbeit mit der "ppc.library" hat (die V7 arbeitet endlich mit der ppc.library zusammen).

Installieren Sie die ppc.library nur, wenn Sie auch die PPC-Turbokarte eingebaut haben! Benutzen Sie möglichst die neuesten Versionen der 68040/68060.library incl. der ppc.library – diese sind unter ftp.phase5.de oder im Aminet verfügbar.

So nebenbei: Es gibt nun Support für die powerpc.library V14, deshalb können Sie sich jetzt für eine Library entscheiden. Grundsätzlich sollte es auch klappen, wenn man die Version unter Frank Wille's ppc.library Emulation für WarpOS (V0.6b oder) laufen lässt.

4. Mehr Informationen über die PPC-Unterstützung und Konfiguration erhalten

Sie unter den entsprechenden FAQ-Seiten - bedenken Sie aber, daß ein Keyfile benötigt wird um alle Funktionen der PPC-Unterstützung innerhalb dieses Programms nutzen zu können.

68020/030+68881/882 FPU und 68040/060+FPU Unterstützung

Die AmigaOS mathieeee-Libraries kümmern sich selbständig um die Unterstützung des Mathe CoProzessors. Allerdings werden diese aus verschiedenen Gründen nicht von diesem Programm genutzt.

- Sie können nicht von verschiedenen Prozessoren geteilt genutzt werden
- Sie sind im Moment unter OS 3.1 nicht optimiert für 68040/060+FPU

Leider unterstützen die genutzten FFP-Libraries die FPU auch nicht.

Aber im Aminet sind verschiedene Patches verfügbar, welche die FPU-Unterstützung grundsätzlich bieten, zusätzliche FPU-Unterstützung den FFP-Libraries ←
hinzuführen,
oder grundsätzlich die 040/060 CPUs besser nutzen, z.B. um die nicht notwendige Emulation von fehlenden Befehlen durch die 68040/68060.library zu umgehen.

Stellen Sie sicher, daß diese Patches nicht mit den verschiedenen Versionen der 680x0 Libraries kollidieren oder vielleicht schon zu diesen gehören. Selbst wenn Sie die Anleitungen sorgfältig gelesen haben, sollten Sie sich trotzdem noch ←
diese

Lösungsmöglichkeiten ansehen:

1. Fehlerbereinigungen innerhalb der Mathe-Libraries:

Dies hat zwar nichts mit den FFP-Libraries zu tun, aber dadurch das auch in der mathieeesingbas.library ein Fehler enthalten ist (welcher sich aber im ROM befindet), sollte sie einen der folgenden Patches dafür installieren:

- a) die beste Lösung dafür ist eine neuere SetPatch Version V43.x (verfügbar von ftp.amiga.de und dort irgendwo unter "/pub/")
- b) sollte SetPatch V43 mit Ihrer OS Version nicht laufen, dann sollten sie z.B. "SetMathPatch" benutzen (ist z.B. in GhostScript enthalten - siehe Aminet:gfx/show)

Diese Patches können Probleme mit einigen Mathe-Library Ersätzen haben - es ist natürlich auch logisch, daß ein komplett neu geschriebener Ersatz der Library in Zukunft auch keinen Patch mehr benötigt.

Nun ja, jedenfalls nicht für die gleichen Fehler...

2. Patchen Sie die Mathe-Libraries für eine bessere (oder einführende) FPU Unterstützung:

- 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) verschiedene andere Patches aus dem "util" Bereich des Aminet

Mit den 68040/68060.libraries von Phase5 - zukünftige Patches der Mathe-Libraries sind nicht mehr notwendig - sollte es aber trotzdem noch funktionieren.

3. Generelle 68040/060 Beschleunigung:

Für eine automatische Erhöhung der Ausführungsgeschwindigkeit auf 68020+ Systemen benutzt dieses Programm die utility.library.

Dieses hat zwar nichts mit der FPU zu tun, aber wenn Sie einen 060 mit OS 3.0 besitzen, dann sollten Sie sich überlegen ob Sie nicht "Mult64Patch" installieren, welches die 64bit Integer Funktionen UMult64/SMult64 der utility.library V39+ (welche auf einem 060 mittels Software emuliert werden muß) ZWEIMAL schneller als das eigentliche Patch der 68060 Library ausführt, und sogar VIERMAL schneller als die Trap Emulation. Ein Testprogramm ist beigelegt.

Das Programm kann im Aminet unter Aminet:util/boot/Mult64Patch.lha gefunden werden - es ist möglich, daß neuere Versionen der 68060.library besser sind. Führen Sie einfach einen Geschwindigkeitstest durch und entscheiden Sie sich dann.

4. Bessere Performance auf 680x0 und PPC:

Die nun folgenden Tools arbeiten sehr gut auf einem 040/PPC Board (in dieser Reihenfolge aus s:startup-sequence entnommen):

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

FastExec V2.9	(Aminet)	-> Verschiedene Geschwindigkeitssteigerungen
SetPatch V43.6b	(www.amiga.de)	-> OS Patches
QuickRom V36.08	(Aminet)	-> ROM nach RAM
CpuBlit98	(Aminet)	-> Läßt die CPU das Blitten erledigen

Diese laufen alle im 60ns Modus, zusammen mit SetFastAvec, PPCInstall und CyberGraphX V3.

1.8 kontakt

Sie können mich wie folgt erreichen:

... per Post/Telefon:

```

|
|      Senden Sie Bug-Report, Geld, etc. an:
|
|-----
|      * SuperView Development & Registration *
|      * DRAFU Development & Registration *
|      * Image Engineer Registration Site Europe *
|
|
|      PerSuaSiVe SoftWorX
|
|      Andreas R. Kleinert
|      Rabenflugstrasse 1
|      D-57074 Siegen
|      Germany, Europe
|
|      +49-271-22869
|      (auch FAX + AB)
|
|      Werktags nach 18.00 Uhr.
|
|      Sie können eine Nachricht hinterlassen, falls
|      ich nicht erreichbar sein sollte - erwarten
|      Sie allerdings keine Rückrufe nach USA,
|      Australien, ... wegen der HOHEN Gebühren.
|
|-----

```

... per InterNet:

Allgemeine PerSuaSiVe SoftWorX WWW Support Site ist

<http://www.ar-kleinert.de>

Binaries bitte nur nach vorherigem Nachfragen und anschließender Bestätigung meinerseits verschicken - mein Postfach ist leider nicht unendlich groß.

```

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

```

1.9 danksagungen

Ein Dankeschön geht an (Auflistung nach Vorkommen ;-)

=====

- Ingo Jürgensmann	- Thomas Boerke	- 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 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. 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 |
| - 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 Surette | - Vincenzo Morra | - Ross Kirk |
| - Michel Verstraeten | - George Elliott | - Kevin Futter |
| - Michael Groni | - Markus Grubinger | - Kimme Utsi |
| - Andrew Baldwin | - Otto Carvalho | - Andreas Krüger |
| - Gerd Schniggenberg | - Luca Ricossa | - Phillip Wright |
-

-
- | | | |
|-----------------------|------------------------|----------------------------|
| - Frédéric Faux | - Elmar Hoffmann | - Jonas Hultén |
| - Johann Samlowski | - Philippe Devilard | - Johan Eriksson |
| - Antonio Brianese | - Michael John | - Rune Jensen |
| - Jürgen Urbanek | - Mikkel Hald | - Hal Samuelson |
| - Norman Caetano | - Per Arne Flø | - Rich Robinson |
| - Adam Corrano | - Beth Hedrick | - Casper Thygesen |
| - Kai Foelster | - Peter Denomy | - Morten Straarup |
| - Thomas Karlsen | - Luca Baldelli | - Leonardo Petrucelli |
| - Richard Gore | - Tom Duin | - Dominique Deangili |
| - Anders Drejer | - Olivier Pertin | - Colin Keefe |
| - Roger Curtis | - Sam Gillies | - Linus Silvander |
| - Klaus-Peter Simon | - Whitford Bates | - Laurent Moussy |
| - Paul Kieffer | - Yves Liebercier | - S. Lichtendahl |
| - Alan Guillevic | - Thomas Lorenz | - Chris Barrow |
| - Ed Eden | - Keith Schyler | - Trond K. Tveit |
| - Janko Köhler | - Andrew Mills | - Howard Toliver |
| - Jon Mines | - Magnus Bouvin | - Dan Muldin |
| - Mahieux Pascal | - James Luscombe | - Peter Mattson |
| - Martin Ruston | - Kapryan Kennedy | - Coeurjoly Fabien |
| - William Eaves | - Cameron Snyder | - Johnny Nielsen |
| - Jason Birnie | - Michael Osmolski | - Kevin Fairhurst |
| - Peter Annuss | - Larry Urquhart | - Philip Yearbury |
| - Neil Bowes | - Steve Hodson | - Johan Rönnblom |
| - Harald Schulz | - Christian Schröpfer | - Michael Fedrowitz |
| - Denis Zwornarz | - Gert Hubers | - Robert Little |
| - Christopher Handley | - Stefano Guidetti | - Jürgen Seubert |
| - Paul Korhonen | - Frank Müller | - Peter Kaltstein |
| - Peter Theuring | - Gunter Kusserow | - Telemar Rosenberger |
| - Phillip Degnan | - John Melville | - Alexandre Kairouannais |
| - Chris Dallimore | - Paul Sadlik | - Matthew Sawyer |
| - Jeffrey Grzanich | - John Hart | - Ian Tyrell |
| - Walt Challenger | - Martin Sprenger | - Pekka Sippola |
| - Brice Terzaghi | - Adrian Cope | - Frank Böhne |
| - Petr Voralek | - Antoine Bordier | - Patrice Dumont |
| - Manfred Kern | - Francis Klein | - Dominique Harelle |
| - Arnljot Arntsen | - Havard Lunde | - Jürgen Ofner |
| - Geoff Tovey | - Herve Sonneviller | - Winek Zawada |
| - Sascha Ploss | - Paul Lang | - David Gerber |
| - Michael Domoney | - Carl Read | - Harald Wuensche |
| - Eirik Synnes | - James Harrison | - Mark Shaw |
| - Frank Wille | - Adam Suwala | - Winfried Krueger |
| - Simon J Glover | - Wolfgang W. Wolber | - Don Cox |
| - Henrik Jensen | - Matteo Consolati | - Jürgen Wilschke |
| - Stephen Webber | - Clive Dennett-Thorpe | - Svein Inge Wik |
| - Philippe Reux | - Paul Venton | - Bjarke Vangsgaard |
| - Rolf Max Rotvel | - Michael Every | - Jason Ruellan |
| - Stefan Fischer | - Roberto Muller | - Michael Thompson |
| - Alfred Kendall | - John Orwin | - Rolf Kleiber |
| - Kari Kortro | - G. Burdett | - Daniel Westerberg |
| - Daniel Stripes | - Scott Konowal | - Steinar Pedersen |
| - William Seaton | - Brian Riis | - Antonio Maria Sebastiani |
| - Dario Soccoli | - Arno Richter | - Chris Appleton |
| - Richard Lane | - Manfred Kern | - Marco van der Laan |
| - Christian Sauer | - Rasmus Bothe | - Neil Mallet |
| - Andreas Ohlsson | - Mark Vallins | - Paul Compton |
| - Craig Peterson | - Gontier Laurent | - Simon Jones |
| - Mathias Roslund | - John de Boni | - Maria Pelova |
-

- | | | |
|----------------------|---------------------|--------------------|
| - Jennifer Symancyk | - David Hibbert | - Bruno Caruso |
| - Wolfgang Bauer | - Michael R. Wilson | - Arsi Koutaniemi |
| - Arthur Moyer | - Thomas Schaefer | - James Miller |
| - Karl-Olav Gravdahl | - Janifer Lopez | - James George |
| - Eric Muller | - Ian Argæt | - Mats-Olov Rustad |
| - Ian Armstrong | - Philip Vedovatti | - Daniel Plant |
| - Klaus-Dieter Klang | - Dirk Pohlmann | - Kevin Hupp |
| - Dietbert Leusmann | - Markus Schmidt | - Frank Aisenbrey |

Dank geht auch an:

- Allen Käufer der SView Productivity Suite von Schatztruhe
- Dem Cloanto Team, namentlich Michael C. Battilana
- Gerd Frank für die Übersetzung und Nachbearbeitung zahlreicher Guides
- Die Mitarbeiter von Phase5, namentlich Ralph Schmidt und Claus Herrmann
- Die Picture Datatypes V43 Programmierer, namentlich Frank Mariak und Olaf Barthel ←
- Die anderen Programmierer von Datatypes, für den Informationsaustausch und die hilfreichen Anmerkungen
- Dutzende von Leuten, an welche ich mich hier leider nicht erinnert habe !

1.10 Datatypes FAQ

OS 3.5: Grundsätzliche Anmerkungen

Grundsätzlich läuft dieses Datatype mit OS 3.5 zusammen. Auch unter der ppc. library (PPC), die powerpc.library (WOS) oder der ppclib-emu läuft es. Mal abgesehen von den Behauptungen, die einige Leute im Usenet stellen. Wie auch immer, manche Leute haben so das eine oder andere Problem, welche aber nichts direkt mit OS 3.5 zu tun haben! ←

So z.B. sollte man unter CyberGraphX V4.2 sicherstellen, daß PLANES2FAST gesetzt ist (bei anderen CGfx Versionen natürlich auch). Manchmal macht es auch Sinn den neuen picture.datatype V44 mit der cgx-basierenden V43 auszutauschen - es ist halt einfach schneller. ←

OS 3.5 Probleme

Programme welche das picture.datatype V44 benutzen um ein On-Screen Dithering durchzuführen, werden mit dem "Problem" konfrontiert, daß 24 Bit Bilder auch bei der Anzeige auf 15/16 Bit Bildschirmen gedithert werden. Bezugnehmend auf das

OS 3.5 Entwicklerteam, sollte dies in einer "bessern Bild Qualität" resultieren.

Bei der Analyse dieser Aussage werden einige Bemerken, daß die meisten auf

PC-Chips basierenden Grafikkarten nur 6 Bit Farbfixierungstabellen (LUTs = color lookup tables) erlauben (das ist, 6 Bit für Rot, Grün und Blau - dies ist nur ein Wertebereich von 0 - 63 anstatt von 0 - 255), welches auch nicht viel besser ist als ein 5:5:5 oder 5:6:5 Wertebereich bei 15/16 Bit Vielfarbmodi. Wie auch immer, 16 Bit Vielfarbmodi erlauben 65536 ↵

unterschiedliche

Farben auf dem Bildschirm. Dies ist darin begründet, das eine 6 Bit LUT nur 256 von 262144 Farben erlaubt.

Wie auch immer, diese neuen V44 Dithering Optionen können mittels der Datatypes Voreinstellungen geändert werden - Die Globalen Einstellungen werden dann lokal überschrieben.

"Nicht genug Speicher"

Der Hauptgrund weshalb dieses Datatype erschaffen wurde, war um ein PPC-Optimiertes PNG Datatype zu erhalten. Die 68K Unterstützung wurde der Vollständigkeit halbre hinzugefügt (und als eine Fallback Option). Wie auch immer, es hat den selben Grundlegenden Aufbau wie auch die PPC Version und diese benötigt auch einiges mehr an Speicher als unbedingt notwendig - dies sollte aber auch die Ladegeschwindigkeit in der 68K Version erhöhen (verglichen mit anderen PNG Datatypes).

Dieses Datatype ist nicht für 2 oder 4 MB Maschinen gedacht - Sie sollten einigen freien Speicher haben - wenn nicht, benutzen Sie einen anderen der verfügbaren PNG-Datatypes.

CTRL-E Unterstützung ?

Nein, nicht auf diesem Weg Kumpel !

Keyfile System

Für dieses Datatype wird ein Keyfile System benutzt - beachten Sie, daß das Keyfile im Moment keine "Extra Funktionen" freischaltet. Es macht die 68K und PPC Module voll Funktionsfähig und ersetzt den Text "Registriert ?" in der Fortschrittsanzeige (die Unregistrierte Version exportiert nur jede 3. Zeile einer Grafiken, was in Streifen resultiert.)

Ich verschicke keinen Keyfile mittels der Briefpost. Wenn Sie Ihr Keyfile erhalten wollen, sollten Sie daran denken Ihre EMail-Adresse (lesbar geschrieben ↵)

mit Ihrer Registration zu schicken !

Bemerkung: Das Keyfile kann in S: plaziert ewrden oder dort wohin KEYPATH (Umgebungsvariable) zeigt.

PPC Modul (WOS)

Dieses Modul ist ein Experiment und befolgt fast die gleichen Regeln wie das Modul für den PPC – es ist einfach nur "akPNG.wos" (190K) genannt und benutzt stattdessen die powerpc.library V14+.

Das Externe Programm "C:LoadElfWOS" wird für die Ausführung der PPC ELF Module benötigt (mit Geschwindigkeitsvorteilen!), solange LOADELF_WOS=OFF in der Voreinstellungsdatei gesetzt ist.

Anmerkungen für LOADELF_WOS=ON:

Vielleicht bringt es eine kleine Beschleunigung wenn man "C:LoadElfWOS" Resident macht (setzen des "p" Bits und ausführen von "Resident C:LoadElfWOS" in s:user-startup). Sie brauchen aber auf jeden Fall eine Version von C:LoadElfWOS welche Resident gemacht werden kann. Vielleicht sollten Sie es einfach einmal ausprobieren.

Anmerkungen für LOADELF_WOS=OFF:

Wenn LOADELF_WOS=OFF gesetzt ist kann es bei einigen Programmen zu Stabilitätsproblemen (*) kommen (z.B. mit dopus_pattern oder WBPatten). Programm spezifische Einstellungen machen hier vielleicht Sinn (z.B. LOADELF_WOS=ON explizit benutzen für diese Programme, aber für andere dann LOADELF_WOS=OFF setzen).

Die Benutzung der CACHE_WOS Option bewirkt, daß das ELF Modul nicht mehr vor jedem Gebrauch wieder in den Speicher geladen werden muß, sondern die ganze Zeit über im Speicher verbleibt (verbraucht zwar doppelt so viel Speicher, insbesondere beim Dekodierungsvorgang, ist aber bemerkbar schneller). CACHE_WOS Einstellungen sollten während der Laufzeit geändert werden.

CACHE_WOS=ON ist zu empfehlen wenn Sie die größte Geschwindigkeit haben möchten und sich nicht über den Speicherverbrauch kümmern brauchen – jedenfalls sollten Sie trotzdem Vorsichtig sein, den wenn Speichermangel auftritt, bricht das PNG-Lademodul ab, was ein noch größeres Problem nach sich ziehen kann. Dies ist ganz besonderst wichtig, weil die WOS Version – für einen kurzen Moment lang – beim Transferieren des Bildes vom PPC auf die 68K Seite, doppelt so viel Speicher verbraucht (dies hat Technische Gründe). (*) Grund ist unbekannt

Abschließende Worte:

Das benutzte ELF Modul für die ppc.library funktioniert natürlich auch mit der aktuellen Beta Version von Frank Wille's ppc.library Emulation für WOS (V0.6b oder höher) – Ich empfehle Ihnen einfach einmal auszuprobieren mit welcher Version es schneller läuft: die WOS Version oder die Emulierte OOC Version. Seitdem die PPC Version "C:LoadElfWOS" nicht mehr benötigt ist dies noch eine offene Frage.

Die aktuellste Version der ppc.library Emulation für WOS finden Sie auf Frank Wille's Homepage unter <http://home.owl.de/~frank/>

Ja, dieses Datatype ist dafür vorbereitet um mit den Phase5 powerUP (TM) Prozessorkarten große Beschleunigungen zu erreichen.

Um dies zu erreichen muß das ELF PNG Dekodierungsmodul nach
SYS:Classes/Datatypes/akPNG.ppc kopiert werden - das Installationsscript erledigt dies für Sie auf Wunsch.

Versichern Sie sich, daß Sie die 68040/060 Versin des Datatypes installiert haben, da die 68000/030 Version den notwendigen Programmcode nicht enthalten (es gibt ja auch keine powerUP Prozessorkarten mit 68000/030 CPUs und geplant sind meines Wissens nach auch keine). Deshalb installieren Sie das ELF Modul und/oder die ppc.library nicht, wenn Sie keine PPC Prozessorkarte eingebaut haben.

Der reine Ladegeschwindigkeit sollte sehr beeindruckend mit den PPC Modul sein. Natürlich kann es die Geschwindigkeit des Anzeigens und Ditherns (remapping) von Systemmodulen oder dem Aufrufenden Programm nicht erhöhen.

Die Umwandlung nach HAM oder das Ordered Dithering (für 24 Bit Grafiken, z.B. wenn man nicht im V43 Modus ist) sind im Moment NICHT optimiert - besorgen Sie sich eine Grafikkarte !

Bitte beachten Sie, daß das Datatype (68K und PPC) nur für Registrierte Benutzer dieses Datatypes, welche das Keyfile installiert haben sollten, die volle Funktionalität bietet.

Sollten Sie kein Keyfile installiert haben, dann haben Sie zwei ↔
Auswahlmöglichkeiten:

1. Entfernen Sie es nochmals
2. Benutzen Sie das 68K oder PPC Module einfach. Aber Sie werden nur jede
3. Bildzeile dabei erhalten (das gesamte Bild wird zwar geladen und umgewandelt, aber nur jede 3. Bildzeile wird weiter gegeben)

Geschwindigkeit: Um die Verarbeitungsgeschwindigkeit des Decodes auszutesten, ↔
sollten Sie

mit AWeb online gehen und eine WWW-Seite mit vielen großen PNG- ↔
Grafiken
laden. Dann gehen Sie wieder offline und laden die selbe Seite ↔
aus dem
Cache: jetzt erhalten Sie einen Eindruck von der reinen ↔
Verarbeitungsgeschwindigkeit
ohne die störenden Einflüsse vom Herunterladen und von anderen ↔
Tasks.

Am besten ist es wenn Sie die Geschwindigkeitstest im V40 Modus ↔
durchführen wenn
Sie die Demoversion nutze. Seit dem V43 Modus haben die ↔
Einschränkungen der Demoversion
(soll heißen, daß ja nicht jede Zeile der Grafik weitergegeben ↔
werden muß) einen
(wenn auch geringen) Einfluß auf die Geschwindigkeit - Die ↔
jeweiligen Zeilen müssen

Extra *gelöscht* werden, was ein bisschen Zeit bei 24 Bit Bildern benötigt.

Entschuldigen Sie - dies entstand aber nach der V44.2 durch einen Fehlerbereinigung.

Anmerkung: Das Dekodieren der Grafik benötigt doppelt so viel Speicher wie bei einem normalen Datatype. Zusätzlich wird noch Speicher für die PPC Version benötigt was für das ELF-Modul 245K beträgt, wobei hier noch weitere 16K für den Stack und 16K für die I/O Puffer benötigt werden (wie Sie wissen, steht RISC für 'Reduzierte Befehlsmenge' und nicht für 'Reduzierter Speicherverbrauch' - aber nun haben Sie wenigstens die Möglichkeit den ganzen teuren Speicher mal zu nutzen ;-)
Zusätzlich ist die Fortschrittsanzeige für das Dekodieren mittels PPC nicht verfügbar (macht auch nicht viel Sinn besonderst wenn man z.B. im WWW unterwegs ist).

Kleines PPC FAQ

F: Warum ist die 060/PPC Kombination schneller als die 040/PPC Kombination?

A: Weil der 060 die I/O Anfragen schneller bearbeiten kann (dazu gehören aus Betriebssystemaufrufe), als der 040. Kleine Unterschiede können auch beim Einsatz verschiedener Festplatten entstehen - um diesen Einfluß zu minimieren kann man z.B. die Dateien nach RAM: kopieren. Aber dies erzeugt keine Ergebnisse wie Sie auch im echten Einsatz vorkommen. Die folgende Frage gehört auch damit zusammen.

F: Kann eine PPC Laderoutine nicht schneller sein, als die bei diesem Datatype?

A: Ja, sie *kann* schneller sein als die ermittelten Ergebnisse anzeigen. Das Problem ist, daß Datatypes mit Bitmaps arbeiten müssen, welche alles ausbremsen. Z.B. im 24 Bit Modus muß die Funktion DTM_WRITEPIXELARRAY auf dem 68K ausgeführt werden und im 8 Bit Modus muß dasselbe für WritePixelLine8() gemacht werden - die späteren Versionen enthalten eventuell eine Funktion für C2P Umwandlungen für Systeme ohne eine Grafikkarte. Hierfür kann man z.B. das alternative PPC Lademodul für die SuperView-Library ausprobieren.

F: Warum gibt es bei verschiedenen Bildern auch verschiedene Beschleunigungswerte?

Ich habe Jan Uerpmann's PicBench Test von seiner InterNet-Seite <<http://www.tu-bs.de/~y0002723/files/PicBench.lha>> getestet, und es scheint dies nur zu bestätigen.

A: Je größer die Bilder sind, desto mehr kann der PPC beim Dekodieren helfen. Jedoch beeinflussen Dateigröße, Bildgröße und Kompressionsfaktor von PNGs diese

Benchmark-Ergebnisse erheblich. Zum Beispiel kann eine kleine Datei mit hohem Kompressionsfaktor mehr für den PPC geeignet sein, als eine große Datei mit niedriger Kompression (bei gleichen Bildgrößen). Größere Bilder können andererseits

bessere Ergebnisse erzielen als kleine Bilder (bei konstantem Kompressionsfaktor).

Der benutzte Benchmark kann dies nicht überprüfen/bestätigen, jedoch haben wir
"durchschnittliche" (und fehlerhafte) Bilder benutzt. ↵

Weitere Datatypes ?

Unter Aminet:util/dtype/ können auch die akTIFF und akJFIF Datatypes gefunden werden. ↵

Keine V43 mit AGA ?

Demnächst erscheint ein V43 picture.datatype welche sich im Picasso96 RTG Archiv (im Aminet) befindet. Dieses Arbeitet auch mit AGA zusammen.

Abstürze?

Der 1. Grund für einen Absturz ist oftmals die Größe des Stacks. Es ist dann nicht genügend vorhanden.

IPrefs/WBPatterns haben diese Problem und oftmals auch andere Programme. Überprüfen Sie dies und/oder die Nutzung von FastIPrefs (der Ersatz dafür) ist zu empfehlen.

Bei anderen Programmen sollten Sie die Stackgröße über das Programmikon vergrößern, oder wenn diese über das CLI/Shell aufgerufen werden (z.B. PPaint), dort die Stackgröße hochsetzen.

Die Benutzung von (Fast)IPrefs im PPC Modus ist eigentlich keine gute Idee, aber bei manchen Personen kann das folgende in s:startup-sequence helfen:

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

Bei anderen funktioniert der Trick aus dem Picasso96 FAQ:
Kopieren Sie den Aufruf des Tools "CPUBlit" (ein alter Patch welcher im Aminet verfügbar ist) in s:startup-sequence *bevor* die Monitortreiber geladen werden. Sie müssen das Tool folgendermaßen aufrufen:

```
CPUBlit -a -b
```

Sie sollten sich außerdem einmal Tools wie FBlit, FastBlit, CpuBlit98 und einige andere ähnliche Tools aus dem Aminet:util/boot ansehen – einige arbeiten perfekt auf Ihrer Maschine, andere vielleicht gar nicht. Aber ein bisschen damit zu experimentieren lohnt sich allemal.

Transparenz (allgemein)

PNG unterstützt Transparenzebenen für jede Farbe aus einem vorgegebenen Bild.

Für Bilder mit zugewiesener Farbpalette wird dies mittels "schattierter" Farbpalette erledigt, welche 0..255 Werte für spezifizierte Farben zur Verfügung stellt:

"0" bedeutet volltransparent, während "255" keinesfalls transparent bedeutet.

Da die Datatypes-Schnittstelle (API) bis V43 für solche Fälle nicht ausgelegt ist, suchen wir einfach nach der ersten Transparenzfarbe, die dem Transparenz-Level "0" gleichkommt. Diese pseudo-transparenten Farben werden immer ignoriert - man kann diese auf einem 256 Farben Bildschirm sowieso nicht wiedergeben.

Sollten Sie also eigene WWW-Seiten mit transparenten PNG Grafiken gestalten: Stellen Sie bitte sicher, daß immer nur eine transparente Farbe benutzt wird, und diese vollständig transparent ist!

Die Nummer der transparenten Farbe ist uninteressant - viele Leute bevorzugen aber trotzdem Farbe #0.

Transparenz für True/High-Color (24bit / 16bit) Bilder (mehr als 256 Farben), z.B. mittels Alphakanal wird momentan nicht unterstützt (und wird es vielleicht auch nie, bzgl. picture.datatype V43)

Transparenz (speziell Browser)

Es gab zahlreiche Fehlerberichte, in denen mir Leute erzählten, daß die Transparenz-Features meistens nicht richtig arbeiteten. Bekanntermaßen enthalten viele Webseiten PNG Grafiken mit einer transparenten Farbe, welche fast gleich der Hintergrundfarbe ist.

Alles was ich dazu sagen kann ist, daß dies nach den mir vorliegenden Informationen, nicht mein Fehler ist.

Die Transparenz-Information als solche wird definitiv richtig gelesen, und es gibt nur einen notwendigen Schritt zu tun - die Weiterleitung an den picture.datatype durch setzen eines speziellen Flags in der BitMapHeader Struktur:

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

Mit dem picture.datatype V43 gab es ehemals auch ein spezielles Flag (mittlerweile veraltet) dafür, aber dies wird nicht benutzt (wurde aber trotzdem getestet).

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

Es scheint, daß weder picture.datatype V39/40 noch V43 dieses Flag richtig interpretieren, egal in welchem Modus (mit oder ohne Remapping).

Theoretisch gibt es zwei Möglichkeiten für ein Programm (z.B. Browser), die Datatype Grafiken zu verwalten:

picture.datatype die Arbeit machen lassen (;)

- Daten laden
- einem Screen/Fenster zuweisen und mitteilen wo sie in welcher Größe erscheinen sollen; Remapping der Screen-Farben erlauben.

=> In diesem Fall hat der picture.datatype die Transparenzverwaltung zu regeln und den Farbpaletteneintrag der Transparenz-Farbe mit dem korrespondierenden Wert der Bildschirmfarben zu ersetzen, *VOR* dem Remapping auf den Bildschirm!

Es ist abhängig vom picture.datatype.

DIY - die Arbeit selbst erledigen

- Daten laden
- ohne Remapping verwerten
- Remapping und Darstellung selbst erledigen, also auch die Transparenz!

=> In diesem Fall wird Transparenz in keinsten Weise vom Datatype abgewickelt ↔
.

Es ist abhängig vom Browser.

Offensichtlich funktionieren beide Wege nicht mit meiner aktuellen Version, obwohl mir gesagt wurde, daß ein anderer Datatype die Arbeit korrekt durchführt. Lustig genug, daß mir dessen Autor erzählt hat er hätte es auf dem gleichen Wege wie ich programmiert...

Gut, alles was ich sagen kann ist: Sendet weiterhin jegliche Fehlerberichte mit obiger Erklärung an euren Browser-Hersteller oder picture.datatype Lieferanten.

Momentan kann ich nicht mehr als das machen, bis mir jemand sagt, wo meine Überlegungen fehlerhaft sind. Aber ich werde nur deswegen sicherlich nicht das OOP Datatypes Konzept aufbohren und das Remapping auf vielleicht vorgegebene Screens selbst richtig durchführen.

Keine Funktion zum Speichern?

Tut mir leid, aber es wird keine Funktion zum Speichern geben (DTM_WRITE Methode ↔).

Dies liegt darin begründet, daß ich der Meinung bin, das Datatypes hauptsächlich ↔ dafür

da sind zum Austausch von Daten und nicht um die Arbeit von schon existierenden Konvertierungsprogrammen zu erfüllen.

Um es ein bisschen genauer zu erklären:

Der Datatype Mechanismus stellt hauptsächlich eine Möglichkeit dar um die Art der Implementation und des Datenformats und aller Details zu verstecken.

Wenn ein Datatype zu viele Möglichkeiten für das Dateiformat der Zieldatei bietet, so ist dies – meiner Meinung nach – komplett gegen dieses Konzept. Der Optimale Weg um das Datatype Konzept sauber zu halten wird sein, alles intern im Amiga eigenen IFF Format zu behandeln – welches z.B. für den Austausch von Daten über das Clipboard von Essentieller Bedeutung ist. Leider ist IFF-ILBM nicht besonders brauchbar für Farbtiefen größer als 8 Bit. Vielleicht wird hierfür IFF-RGFX eine gute Wahl sein...

Die bescheidene Bildschirmmodus Auswahl

Die Funktion BestModeID aus der graphics.library ist nicht besonders gut erstellt worden. Versuchen Sie mittels eines Patchen auf eine bessere Version umzusteigen, z.B. Aminet:util/sys/ModeP.lha

Fortschrittsanzeige und Programme (insbesondere Browser)

Bitte beachten Sie, daß die (optionale) Fortschrittsanzeige sich auf einem Bildschirm öffnet, welcher mittels pr_WindowPtr spezifiziert wurde, oder halt auf dem als Standard definierten Public Screen. Insbesondere dann, wenn Ihr bevorzugter Browser pr_WindowPtr nicht setzt, oder seinen Bildschirm nicht als den Standard Public Screen deklariert. Das liegt aber nicht an mir. PDTA_Screen wird zwar als erstes geprüft – aber dies funktioniert auch nicht immer.

Ramlib Abstürze

Wenn Sie "ramlib" Gurus innerhalb eines anderen Programmes erhalten sollten, dann installieren Sie doch einmal versuchsweise Aminet:util/sys/StackAid.lha

Probleme mit IBrowse

Wenn sie die internen (68k) Laderoutinen von IBrowse umgehen und stattdessen (PPC) Datatypes verwenden wollen, treten manchmal obskure Probleme auf (es werden nur teilweise Grafiken dekodiert und angezeigt).

Dies scheint an einer falschen Prioritäteneinstellung zu liegen, ob interne oder externe Dekoder verwendet werden (und/oder an MCP ;).

Wenn Sie wirklich Datatypes fürs dekodieren verwenden wollen, sollten Sie dessen Prioritäten z.B. auf 10, die der anderen auf 0 setzen.

Fehler wurde zuerst gemeldet von Boris Bojic <bbojic@arco.met.fu-berlin.de>

Warum werden "progressive" Bilder nicht progressiv dargestellt ?

Weil die Programmierschnittstelle (API) des picture.datatype bis V43

komplett darauf aufbaut, daß Datatypes der Unterklasse "picture" komplette Bitmaps zurückgeben.

Dummerweise können Datatypes nicht: (um nur einige mögliche Lösungsvorschläge zu geben)

- viele kleine Bitmaps (eine für jede Zeile) verwalten
- während des Lesens einer Bilddatei die Kontrolle an den picture.datatype zurückgeben
- in eine existierende, vorgebene Bitmap schreiben

Deshalb gibt es momentan keine Möglichkeit Bilder progressiv darzustellen.

Übrigens wäre die progressive Anzeige im PPC-Modus sowieso eine dumme Idee...

Unbekannte Datatypes (V43)

Wenn Ihr Datatype aufhört zu laufen (unbekanntes Dateiformat), dann schieben Sie nicht gleich die Schuld auf mich, sondern prüfen als erstes ob Sie nicht eine schon längst veraltete Betaversion des picture.datatype V43 benutzen...

Und versichern Sie sich, daß sie nicht das picdtpatch (v39.2), welches aus dem Hypertext.Datatype Archiv von Stefan Ruppert stammt, benutzen.

1.11 Über PNG - den Nachfolger zu GIF

PNG ist der Nachfolger des GIF Dateiformats. Anders als GIF ist es völlig frei von Patenten Dritter und wurde für den freien Datenaustausch geschaffen.

Lasst GIF fallen für PNG! Freie Algorithmen sind genauso wichtig wie Redefreiheit im InterNet!

GIF ist veraltet - Sie sollten es weder weiterhin nutzen noch unterstützen!
Wenn Sie WWW Design machen, nutzen Sie stattdessen PNG oder JPEG.
Es ist wichtig!

Weitere Informationen zu PNG (gesprochen: PiNG) finden Sie z.B. unter:

- [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.

1.12 PNG-Box - Ein InterNet-Tool zum Schreiben von PNG

PNG-Box

© 1997 - 2000 by Andreas Ralph Kleinert.
Alle Rechte vorbehalten.

EIN PerSuaSiVe SoftWorX PRODUKT.

- SHAREWARE -

Programm Information

Ab sofort können Sie ganz einfach zu PNG wechseln!

Es liest jedes von der SuperView-Library unterstützte Grafikformat und kann daraus eine PNG-Datei fürs WWW machen.

Natürlich mit einigen spezifischen Optionen:

- Progressiv ein/aus
- Transparenz ein/aus (und setzen der Farbe 0 .. Max. AnzahlFarben)
- Kompression 0..9

Die Programmoberfläche zeigt einige nützliche Informationen, z.B. Effizienz der Kompression (in Bytes) oder die bereits erledigte Arbeit.

Es wurde nach StyleGuide-Richtlinien programmiert und nutzt die wizard.library ↔
.

Es ist ebenso möglich die Quell- und Zieldateien in einem Workbench-Fenster zwecks Vorschau anzuzeigen.

Es gibt PPC/WOS optimierte Speicher-Module (nur registrierte Version) und nutzt ebenso die Vorteile der PPC/WOS Lade-Module der superview.library.

Informieren Sie sich im Programmarchiv über weitere Angaben zu Copyright und Distribution, z.B. Download über AmiNet:gfx/conv/PNG-Bix.lha

1.13 versionsgeschichte

Versionsgeschichte

=====

- V44.112 (14.8.2001): - Optimierungen
- verwendet nicht länger die utility.library
- V44.110 (3.8.2001): - noch schneller und kleiner
- Version des PPC-Plugins war nicht aktuell (jetzt .110)
- ausserdem wurde PROGRESSBAR-Unterstützung entfernt (noch ↔ ein paar KB)
- V44.100 (1.8.2001): - Optimierungen für 68k und PPC:
- PPC-Version nur noch ca. 123KB
- 68k-Version nur noch ca. 90-99KB
- Upgrade auf libpng 1.0.12 (68k und PPC)
- V44.98 (13.5.2001): - IO-Beschleunigung für 68k-Version
- Memory-Pools für 68k und PPC Hilfscodes
- Upgrade auf libpng 1.0.11
- V44.97 (24.3.2001): - downgrade der PPC-Version auf libpng 1.0.8,
da irgendetwas in 1.0.9+ einen Compilerbug
zu aktivieren scheint
- neues Installer-Skript von Robert C. Reiswig beigefügt
- das alte Problem mit dem Abbruch-bei-Fehler-Bug
korrigiert, bei dem der Stackswap nicht wieder rückgängig
gemacht wurde (-> Laire)
- V44.96 (18.3.2001): - Upgrade auf libpng 1.0.10ß1
- Speed-Optimierung (sowohl 68k als auch PPC)
- keine Crashes mehr
- V44.95 (10.2.2001): - Upgrade auf libpng 1.0.9
- Speed-Optimierung (sowohl 68k als auch PPC)
- Unterstützung für WOS-Crap entfernt; benutzen Sie ppplibemu ↔
...
- V44.90 (3.2.2001): - Upgrade auf libpng 1.0.8
- Speed-Optimierung (sowohl 68k als auch PPC, jedoch nicht ↔ WOS)
- V44.82 (8.8.2000): - SPEEDUP-Option deaktiviert (geht jetzt nicht mehr)
- Problem mit Interleaved-Bildern korrigiert
- kleinere Speedups der PPC/WOS-Versionen
- V44.81 (7.8.2000): - genereller Speedup aller 3 Versionen (68k/PPC/WOS),
besonders bei kleineren Dateien
- leichter Speedup der unregistrierten Version
- Speedup der registrierten Version
- V44.80 (23.7.2000): - Upgrade auf libpng 1.0.7
- V44.79 (13.5.2000): - Upgrade auf libpng 1.0.6
- deutsche Übersetzung hinzugefügt
- Aminet-Release
-

Wegen Faulheit des Übersetzers wurde alles vor V44.79 (30.04.00) nicht ins Deutsche übersetzt... ;^) (Umfang gut 500 Zeilen!)

1.14 bedienung

akPNGPrefs

akPNGPrefs ist das Voreinstellungsprogramm für das akPNG.datatype.

Die Oberfläche wurde mit StormWizard v2.0 entwickelt. Dadurch benötigt dieses Programm die "wizard.library" V37+ (welche Sie im AMINET unter "biz/haage/WizardLibrary.lha" finden können, oder eventuell sogar eine neuere Version unter ftp.haage-partner.com).

Das Piktogramm stammt von Bert Bosma <lmb@wxs.nl> (basierend auf NewIcons).

Ein MUI-Ersatz des Voreinstellungsprogrammes von Alvaro Thompson (das Original) und Achim Stegemann (die späteren Versionen) ist nun unter util/dtype/akMUIPrefs.lha verfügbar - dort sind außerdem noch einige andere Ersatzprogramme zu finden.

Die globalen Einstellungen werden unter ENV: (außerdem auch noch unter ENVARC:) in einer Voreinstellungsdatei namens "Datatypes/akPNG.prefs" gespeichert.

Task-spezifische Einstellungen können auch erstellt werden - mittels des Voreinstellungsprogramm (welches eine Auswahl des jeweiligen Prozesses aus einer Liste ermöglicht, solange dieses zur selben Zeit läuft), oder per Hand gemäß dem folgenden Aufbau:

OPTIONAL

----- Task-Spezifische Einstellungsdateien -----

Voreinstellungen für die verschiedenen aufrufenden Programme können durch das Kopieren der globalen Voreinstellungen ("ENV:Datatypes/akPNG.prefs") zu einer Task-Spezifischen Voreinstellungsdatei getätigt werden.

"ENV:Datatypes/akPNG.prefs_Tasks/TaSkNaMe"

Wobei "TaSkNaMe" den Namen des Programmes darstellt, welches z.B. mittels eines System-Monitors ermittelt werden kann. Dies funktioniert sehr gut mit Workbench-Programmen, welche keine Namensteile benötigen, wie einige CLI Programme. Als Beispiel "CLI(3):Work:Browsers/XWebber". Für z.B. AWeb müssen Sie nur Ihre Datei mit den globalen Voreinstellungen ändern und wie folgt vorgehen:

```
MakeDir ENV:Datatypes/akPNG.prefs_Tasks
```

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

[... und das gleiche für ENVARC: ...]

Danach wird AWeb die globalen Voreinstellungen ignorieren und seine eigenen aus der angegebenen Datei benutzen!

Sie haben die folgenden Einstellungsmöglichkeiten (rechts die Standardvorgaben):

1) V43MODE=(256 24BIT)	(24BIT)
2) V40_24BIT_MODE=(DITHER_ORDERED HAM_OUTPUT)	(DITHER_ORDERED)
3) V40_DEPTH=(3..8)	(8)
4) HAM_MODE=(HAM6 HAM8)	(HAM6)
5) INTERLEAVED_BM8=(ON OFF)	(ON)
6) DISPLAYABLE_BM8=(ONF OFF)	(OFF)
7) PROGRESSBAR=(ON OFF)	(ON)
8) CUSTOM_MODES	---
9) PPC=(ON OFF)	---
10) AUTO=(ON OFF)	(ON)
11) PPCLIB_EMU=(IGNORE USE)	(IGNORE)
12) CACHEWOS=(ON OFF)	(ON)
13) LOADELF_WOS=(ON OFF)	(ON)
14) NOASPECT	---
15) DEBUG	---
16) V44_DITHER=(0..2)	(1)

Grundsätzliche Erklärung der Optionen

=====

1) V43_MODE

NO_DITHERING: Bei Benutzung der pic-dt V43 wird eine 24 Bit Ausgabe geliefert
 V40_DITHERING: Bei Benutzung der pic-dt V43 wird in den V40 Modus gewechselt

2) V40_24BIT_MODE (bei Benutzung des picture datatype V40 oder V43 im V40 Modus)

DITHER_ORDERED: Führt ein "ordered dithering" der 24 Bit Daten durch
 HAM_OUTPUT: Konvertiert die 24 Bit Daten nach HAM6/8

3) V40_DEPTH

Wenn eine vorhandene Farbpalette benutzt wird (z.B. wenn der V40 Modus eingesetzt wird und Ordered Dithering ausgewählt ist), so wird die Anzahl der vorhandenen Farben in der Palette reduziert werden (z.B. auf ECS-Systemen),

wobei 256 Farben der Standard sind. Gültige Farbtiefen sind 3..8 (welche in 16..256 Farben resultieren, was einfach mittels der Formel $2^{\text{Farbtiefe}}$ errechnet werden kann).

4) HAM_MODE (nur mit "HAM_OUTPUT")

HAM6: Erzeugt HAM6 Ausgabe (max. 4096 verschiedene Farben auf dem Bildschirm)

HAM8: Erzeugt HAM8 Ausgabe (über 262144 verschiedene Farben auf dem Bildschirm)

Im HAM8 Modus werden Graustufenbilder mit 256 Graustufen exportiert (sieht schlimm aus in HAM6 und AGA-Rechner können nur 256 Graustufen verwalten ←)

Beachten Sie bitte, das HAM8 nur bei AGA Maschinen verfügbar ist und dadurch Schwierigkeiten mit Grafikkarten erzeugen kann und nicht auf OCS/ECS Amigas laufen wird.

Bei HAM6 und Grafikkarten können auch Probleme auftreten.

5) INTERLEAVED_BM8

ON: Erzeugt Interleaved Bitmaps mit bis zu 256 Farben

OFF: Erzeugt normale Bitmaps (BMF_CLEAR und eventuell BMF_INTERLEAVED) - Sie sollten für einige Programme auf ON wechseln, insbesondere dann wenn dies nicht mit diesen Bitmaps umgehen können, oder wenn AllocBitmap() gepatcht für Chunky Modi z.B. durch die Grafikkartensoftware oder EGSPlus.

6) DISPLAYABLE_BM8

ON: Erzeugt Anzeigbare Bitmaps mit bis zu 256 Farben

OFF: Erzeugt normale Bitmaps (BMF_CLEAR und eventuell BMF_INTERLEAVED) - Sie sollten für einige Programme auf ON wechseln, insbesondere dann wenn diese die vom Datatype erzeugten Bitmaps direkt als eine Screen Bitmap verwenden wollen. Wenn diese Programme diese Möglichkeit bieten, so spart dies einiges an Speicher (für andere Bitmaps). Dies ist besonders für Systeme ohne Grafikkarte und wenig ChipMem zu empfehlen.

7) PROGRESSBAR

ON: Zeigt die Fortschrittsanzeige mit Prozentangabe an

OFF: Zeigt die Fortschrittsanzeige mit Prozentangabe nicht an

8) CUSTOM_MODES (versteckte Option)

Wenn das Schlüsselwort CUSTOM_MODES benutzt wird, werden nur Anzeigemodi aus dem Standardset erzeugt:

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

Wenn CUSTOM_MODES=0x##### (z.B. CUSTOM_MODES=0x00000000) gesetzt ist, wird immer die angegebene hexadezimale ID des Anzeigemodus benutzt - alternativ dazu kann man den Namen des Anzeigemodus angeben, z.B. "CUSTOM_MODES=PAL:HighRes".

Beachten Sie bitte, daß die Schreibweise hier sehr wichtig ist!

Die Anzeige im HAM Modus ist nur möglich, wenn der benutzte Anzeigemodus auch die Möglichkeit bietet HAM-Grafiken anzuzeigen (dies wird durch ein "ODER'n" der Modus-ID mit dem HAM_KEY ermittelt). Wenn dies nicht möglich ist, wird ein ← anderer Anzeigemodus erzeugt.

9) PPC (versteckte Option)

ON: Sollte das PPC- oder WOS-Modul installiert sein, so wird es jetzt genutzt.
 OFF: Wenn die Option PPC=OFF gesetzt ist, so wird das PPC-Modul nicht genutzt, auch wenn ein PPC verfügbar ist. Das Datatype schaltet dafür in den 68K- ← Modus zurück. Dies ist z.B. sehr nützlich beim Vergleichen der Geschwindigkeit.

Dies ist eine Optionen welche nur WÄHREND der Laufzeit zur Verfügung steht. AUTO und PPCLIB_EMU werden immer ausgeführt.

10) AUTO

ON: Versucht herauszufinden, welcher PPC-Kernel installiert ist.
 OFF: Geht einfach davon aus, daß es die ppc.library ist...

Mit AUTO=OFF wird nicht versucht die powerpc.library zu öffnen.
 Dies verursacht möglicherweise Fehler, wenn die V14+ installiert und aktiviert sein sollte. Dies kann auch geschehen, wenn Sie eine PPCLib Emulation laufen ← haben.

11) PPCLIB_EMU

IGNORE: Mit AUTO=ON und installiertem WOS wird die WOS Version genutzt
 USE: Mit AUTO=ON und installiertem WOS wird die PPCLib Emulation genutzt

Natürlich gilt dies nur für die powerpc.library V14+ von WarpOS.

12) CACHE_WOS

Diese Option wird in den FAQ erklärt.

13) LOADELF_WOS

ON: Es wird jetzt "C:LoadElfWOS" anstatt der internen ELF Laderoutine benutzt, um einige Probleme z.B. mit der DOpus Anzeigefunktion oder dem DOpus/Workbench Hintergrundpattern Tool zu umgehen. Bitte benutzen Sie nicht noch zusätzlich CACHE_WOS, dadurch würde Speicher unnötig verschwendet werden.

OFF: Die interne ELF Laderoutine wird jetzt benutzt. Die Nutzung von CACHE_WOS macht nun Sinn. Wenn Probleme mit dieser Option auftreten sollten, dann versuchen Sie zuerst einmal den Stack für den Aufruf dieser Anwendung hochzusetzen. (z.B. durch Erhöhen des Stacks von MultiView auf 32768 Bytes innerhalb des Icons).

14) NOASPECT (versteckte Option)

Wenn das Berechnen des X/Y Aspekts Fehler erzeugen sollte, z.B. bei PictIcon, wird empfohlen diese Option zu benutzen um den 1:1 Aspekt zu erzwingen.

15) DEBUG (versteckte Option)

Aktiviert die Ausgabe von Debug-Meldungen, z.B. durch Öffnen eines Informationsrequesters in welchem detaillierte Angaben über die Bildgröße und die Kompression enthalten sind.

Im 68K Modus wird zusätzlich ein Requester geöffnet, wenn beim Umwandeln des Bildes Fehler aufgetreten sind. Im PPC Modus werden diese Fehler ignoriert.

DIESE OPTION IST NOCH NICHT IMPLEMENTIERT!