

**PNG**

Copyright © CopyrightÂ©1995 Cloanto Italia srl

<b>COLLABORATORS</b>
----------------------

	TITLE :  PNG		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY		November 20, 2024	

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>PNG</b>	<b>1</b>
1.1	PNG DataType . . . . .	1
1.2	Installation . . . . .	2
1.3	The PNG Format . . . . .	2
1.4	PNG Letter . . . . .	3
1.5	From GIF to PNG . . . . .	5
1.6	PNG Specification . . . . .	17
1.7	GIF to PNG ARexx Script . . . . .	17
1.8	GIF to PNG ARexx Script - Comments . . . . .	19
1.9	DataTypes . . . . .	20
1.10	C Source Code . . . . .	21
1.11	Personal Paint . . . . .	28
1.12	Orders . . . . .	30
1.13	Distributors . . . . .	30
1.14	Legal Notes . . . . .	31
1.15	Cloanto . . . . .	33

# Chapter 1

## PNG

### 1.1 PNG Data Type

PNG and the PNG Data Type v. 1.0, by Cloanto <sup>1</sup>

When we heard news that several, very popular, graphics formats were menaced by software patents, at Cloanto we took a strong position in favor of a new, improved format: PNG (Portable Network Graphics).

We are presenting here a PNG Data Type for the Amiga, some background information on PNG, sample source code to use Data Types and an ARexx script to automatically find GIFs and convert them to PNG. As the creators of Personal Paint, we've also included some information on that program.

The Data Type supports all types of PNG-encoded images. Grayscale images are converted to palette-based images. True color (up to 48-bit) images are dithered using a fast Floyd-Steinberg method. (More professional color quantization options are available in Personal Paint.)

This Data Type also reads the Author, Copyright and Comment fields (DTA\_ObjAuthor, DTA\_ObjCopyright and DTA\_ObjAnnotation Data Type attributes). The included source code shows how to read this information.

The Legal Notes contain a distribution license and other information.

Your feedback is very appreciated.

For more information:

Installation	PNG	Data Types
Personal Paint	Cloanto	Legal Notes

<sup>1</sup> The material presented here and the name Cloanto are respectively Copyright © 1995 and a registered trademark of Cloanto Italia srl.

---

## 1.2 Installation

### Installation

The PNG Data Type can be installed with a double-click on the included Installer icon. This automatic installation procedure uses the Amiga Installer program, which must be stored in the SYS:Utilities directory of your system. Manual installation is described in the following lines.

The PNG Data Type can be installed by copying PNG to DEVS:DataTypes, and png.datatype to SYS:Classes/DataTypes. If the files are copied using Shell commands, the PNG icon file ("PNG.info") must also be copied manually (to DEVS:DataTypes).

Users of Amiga systems with a 68020 or higher CPU can install a special version of the Data Type, written to take advantage of new features of these more powerful CPUs. This file is named "png.datatype.020", and must be renamed to "png.datatype" once installed SYS:Classes/DataTypes. (This is also taken care of by the automatic installation script.)

For more information:

PNG

DataTypes

## 1.3 The PNG Format

### The PNG Format

The PNG (Portable Network Graphics) file format was designed as a replacement and extension to GIF and LZW-based TIFF, after Unisys Corporation began demanding royalties on GIF/LZW code.

PNG is gaining general recognition as the best lossless format for storing digital images. (JPEG remains recommended for storing real world images where minimum storage occupation is a priority and loss of information is acceptable.) PNG has the potential to replace both GIF and TIFF, and as a unifying force it should attract more attention beyond that.

For more information:

PNG Letter

Article

PNG Specification

GIF to PNG ARexx script

---

## 1.4 PNG Letter

### Graphics Community Endorses a New File Format

May 1, 1995. A coalition of major software developers, publishers and technical writers announced today its endorsement for the new PNG graphics format. PNG (Portable Network Graphics, pronounced "ping") is a flexible and open format for storing bitmapped graphics images. This effort began in late 1994, when CompuServe and Unisys stunned the online world by announcing that royalties would be required on the formerly freely used GIF file format.

Several companies claim a patent on the LZW compression algorithm, which is an integral part of the GIF file format. Unisys is now requiring developers, publishers, and vendors to pay royalties on any software that either creates or displays GIF files. In response to this announcement, developers hastened to replace the GIF file format with an improved royalty-free format. A coalition of experienced independent graphics developers from the Internet and CompuServe formed a working group and proceeded to design the new format. The result is the PNG format.

PNG is a major advance over the venerable GIF format. By adopting PNG, you would not only be helping the computer graphics community free itself from the Unisys patent, but you would be enjoying the advantages of a powerful new graphics file format. Converting your GIF collections to PNG offers the following benefits:

- \* PNG retains GIF's strength as a simple and portable graphics format.
- \* PNG's compression method has been thoroughly researched and judged free from patent problems.
- \* PNG allows support for true color and alpha channel storage. Its extensible structure leaves room for future requirements.
- \* PNG's feature set allows conversion of all GIF files.
- \* On average, PNG files are smaller than GIF files.
- \* PNG offers a new, more visually appealing, method for progressive display than the scanline interlacing used by GIF.
- \* PNG is designed to support full file integrity checking as well as simple, quick detection of common transmission errors.
- \* Implementations of PNG are royalty-free.

The advantages of making PNG an industry-standard file format are clear. We are now presented with a rare opportunity to move forward in the area of royalty-free graphics display and archiving software. Please help with the adoption of PNG by supporting it as your preferred graphics file format. For more information, source code, file specifications, developer tools, and freeware file converters, you can contact the comp.graphics Internet newsgroups or the Graphics Support Forum on CompuServe (GO GRAPHSUP). For files, check the ftp.uu.net:/graphics/png directory, or email png-info@uunet.uu.net.

---

Thank you for supporting this project.

Signed by:

Michael Abrash, author, Zen of Graphics Programming  
Michael Console Battilana, Cloanto (Personal Paint/Write, etc.)  
Bradley Bell & Elizabeth Piegari, TriSoft (Depth Dwellers)  
Andrei Belogortseff, ChaoSoft (FM StepUp, FM Toolbar, FM Guard, etc.)  
C. Steven Blackwood, Cytherean Adventures (Cargo Bay)  
Robert K. Blaine, ECONO-SOFT  
John Bradley, author of XV  
John Bridges, author of GRASP, PC Paint and PICEM  
Rick Byrnes, The Software Development Group (NoteWorthy, MoneyWise, Eventz, and various shareware products.)  
Tony Caine, ARCAINE Technology  
George Campbell, OsoSoft (Winclip, etc.)  
Mike Ceranski, President, Dvorak Development  
Lee Crocker (Piclab, PGIF, GTools)  
Karen Crowther, Redwood Games (Math Rescue, Word Rescue, Pickle Wars)  
E. Nicholas Cupery, Farba Research (Farba Utilities (tm))  
Thomas Boutell, author of the gd library and the World Wide Web FAQ  
Gary Elfring, Elfring Soft Fonts (Clip Art)  
Steve Estvanik, Cascoly Software (Winzle, Windows in Time, MVP Bridge)  
Jim Faliveno, Monumental Computer Applications, Inc. (TagVue-CaddView)  
Dan Farmer, POV-Team (POV-Ray)  
Oliver Fromme, TBH-Softworx (QPEG, PicDex)  
John Gallant, First Magnitude (3-Ball Juggler, Beat the Bomb, Math Sampler)  
Lawrence Gozum, author (VIDVUE)  
Phil Grenetz, Ivden Technologies  
Diana Gruber, Ted Gruber Software, Inc. (Fastgraph)  
David Hofmann (Computer Graphics Artist, Germany)  
Michael D. Jones, Insight Software Solutions (Finance/Hobbies/Word Games)  
Lutz Kretzschmar, coauthor of Ray Tracing Worlds (Moray)  
Tom Lane, organizer, Independent JPEG Group (IJG JPEG software)  
Steve Lee (Atlantic Coast plc)  
Ralph Mariano @ STReport International Online Magazine  
David K. Mason, author of Morphing on Your PC, coauthor of Making Movies on Your PC (DTA, DFV, DMorf)  
Randy Maclean, Formgen Corp.  
Brad McLane, Caladonia Systems Inc. (Code.Print, ToolThings)  
Al Meadows/Fineware Systems (Author of Space Hound, Peeper, etc.)  
Scott Miller, Apogee Software, Sultans of Shareware  
Jeff Napier, Another Company (Computer Magic)  
Peter Nielsen, Raja Thiagarajan, Julie England (PMView & PMSnap for OS/2)  
David Noakes, Fugue Software  
Dick Oliver, author of PC graphics books and software including Tricks of the Graphics Gurus, PC Graphics Unleashed, and FractalVision  
Dan Richardson, illustrator, author of Create Stereograms on Your PC  
John Richardson, Rogue Marketing (Amazing Secrets Series, Gambling Secrets, JobDisk)  
Steve Rimmer, Alchemy Mindworks Inc. (Graphic Workshop, etc.)  
Greg Roelofs, Info-ZIP (Zip, UnZip and related utilities)  
Guy Eric Schalnatz, Group 42 (PNGLIB, GraphX Viewer)  
Paul Schmidt, Photodex Corporation, GDS (The Graphics Display System)  
Monty Shelton, CrystalWorks (EZCosmos, SIRDS for NIRDS, Language Wiz)  
Steve Sneed, Ozarks West Software, Inc. (OzCIS, OzWin, OZBEXT/OZGIF)

---



David Snyder, MVP Software (MVP Paint)  
Chuck Steenburgh, Tay-Jee Software (Palantir for DOS & Windows, S.O.S.)  
Peter Tiemann (author of TrueBase)  
Glen Tippetts, NeoSoft Corporation (NeoPaint, NeoBook, etc.)  
Rod Underhill, Computer Fine Artist (CIS Comic Forum's Underhill Gallery)  
John Wagner (Improces)  
Bruce F. Webster, Pages Software Inc (WebPages by Pages)  
Tim Wegner, author of Image Lab and Fractal Creations (Fractint)  
Rosemary West, R. K. West Consulting (By The Numbers, LoveDOS, etc.)  
Thomas R. White, Recreational Engineering Associates (MultiMedia Swiss Army Knife)  
Charles L. Wiedemann, REXXcom Systems (XL2001, E-Z-Book, etc.)  
Terry Wilkinson, CIO, AffNet Publishing  
Ben Williams, Black Belt Systems Inc. (WinImages, Imagemaster, etc.)  
Jeff Woods, deltaComm Development, Inc. (Telix for Windows)

-----  
PLEASE COPY AND DISTRIBUTE WIDELY  
-----

## 1.5 From GIF to PNG

The GIF Controversy: A Software Developer's Perspective

January 27, 1995 - Text revision 2 - March 31, 1995  
Copyright © 1995 Cloanto Italia srl, All rights reserved  
Parts are quoted with permission from CompuServe Information Service  
Parts are excerpted from the PNG specification

This article was written with great care. It may reflect personal opinions of the author, which are not necessarily shared by the publishers, who cannot assume any responsibility for mistakes or misprints. Nothing in this article should be regarded as legal counsel. If you require legal or other expert assistance, you should consult a professional advisor. Many of the designations used by manufacturers and sellers to distinguish their products are trademarks. The author of this article has made every attempt to supply trademark information about manufacturers and their products. GIF and Graphics Interchange Format are service marks of CompuServe Inc., an H&R Block Company. PostScript is a registered trademark of Adobe Systems Inc. TIFF is a trademark of Aldus Corp.

Abstract

Introduction

Software Patents

10 Years of LZW

December 29, 1994 - The Days After

Reference

---

## Bibliography

## Abstract

During the past eight years, GIF (Graphics Interchange Format) peacefully became the most popular file format for archiving and exchanging computer images. At the end of December 1994, CompuServe Inc. and Unisys Corporation announced to the public that developers would have to pay a license fee in order to continue to use technology patented by Unisys in certain categories of software supporting the GIF format. These first statements caused immediate reactions and some confusion. As a longer term consequence, it appears likely that GIF will be replaced and extended by the new PNG (Portable Network Graphics) format.

## Introduction

This is a very interesting case, which could teach more than one lesson on the theory and practice of software and the laws. There are many entities involved. Fingers have been pointed at lawmakers, Unisys, CompuServe and developers. In theory, it may have been possible for any or all of these parts to prevent the matter from creating so much anxiety in the first place. Yet we are all here, debating on this issue. This article intends to provide a collection of information from the history of the controversy to the most recent events, as they were perceived by a software developer.

CompuServe released GIF as a free and open specification in 1987. GIF soon became a world standard, and also played an important role in the Internet community. It was well supported by CompuServe's Information Service, but many developers wrote (or acquired under license) software supporting GIF without even needing to know that a company named CompuServe existed. GIF was relatively simple, and very well documented in books, articles and text files.

GIF images are compressed to reduce the file size. The technique used to compress the image data is called LZW (after Lempel-Ziv-Welch) and was first described by Terry A. Welch in the June 1984 issue of IEEE's Computer magazine. Unisys holds a patent on the procedure described in the article, but the article describing the algorithm had no mention of this. The LZW procedure was simple and very well described, and it soon became a very popular technique for data compression (just as GIF would become a standard in its own field). It appears that neither CompuServe, nor the CompuServe Associate who designed GIF, nor the computer world in general were aware of the patent. GIF is not alone in the use of LZW. The TIFF file specification also includes LZW-compression among its compression methods, and so do dozens of very popular file archiving programs (such as Compress).

While having the right to pursue legal action or seek damages against infringing LZW developers and publishers, Unisys has so far been very accomodating and fair. It is likely that the success of LZW and its thousands of implementations, especially among small developers, caught Unisys unprepared. Otherwise, it would be difficult to understand how

---

Unisys could first allow a very large number of small and big developers to use LZW for years, and then, after the establishment of various standards based on LZW, change its attitude.

The original CompuServe/Unisys licensing agreement text which had upset so many developers was immediately followed by clarifications from both CompuServe and Unisys. Given that the online community tends to be suspicious about anything that is big, has a legal department or owns software patents, Unisys had to face a particularly delicate challenge. But it probably wasn't easier for CompuServe, who had to explain the patent issue to its own developers, some of whom felt "betrayed". The outside world would learn about this issue from the press in the following days.

Even Time Magazine reported about this matter, although like most of the newspapers it concentrated on GIF more than on TIFF, LZW, Unisys or software patents. In the meantime, a group of leaders of the online graphics community began working on a patent-free future of GIF. These efforts would later converge into the PNG specification. The full texts of official statements from CompuServe and Unisys are also included at the end of this article (see Reference ).

Among the first reactions, some bulletin board systems had all GIF files deleted from their hard disks (or converted into JPEG format). Common remarks included:

"PROTEST OF NEW COMPUSERVE-UNISYS GIF USAGE TAX !!"

"They [CompuServe] seem to think that GIF is the greatest thing since free online magazines."

"The announcement by CompuServe and Unisys that users of the GIF image format must register by January 10 and pay a royalty or face lawsuits for their past usage, is the online communications community's equivalent of the sneak attack at Pearl Harbor."

These reactions may require some clarification.

Unisys, and not CompuServe, has been "trying to impose" a royalty. The problem is not specific to GIF, but includes TIFF and archiving software.

GIF files are not covered by the patent. There is no risk in distributing GIF files or in using the GIF name. According to a CompuServe spokesperson,

"Recent discussions of GIF taxes and fees are totally without merit. For people who view GIF images, who keep GIF images on servers, or who are creating GIF images for distribution, the recent licensing discussions have no effect on their activities."

Only the software employing the LZW algorithm for writing GIF files is "at risk". The Unisys patent includes claims which specifically cover the decompression of LZW-compressed material, so it may also affect simple GIF readers. Several patent attorneys consulted on this matter have concluded that decompression-only programs do not infringe upon the Unisys patent. Unisys however does not appear to share this opinion.

---

A format such as JPEG cannot be used as a substitute for GIF. Unlike GIF (and PNG), JPEG was designed as a "lossy" format. This means that it slightly changes an image as it is compressed. This is unacceptable for many applications. Also, while JPEG excels in compressing real world true color images, it offers no support for palette-based images.

The CompuServe licensing agreement was intended as a voluntary service to the few dozen developers creating software for use primarily in conjunction with the CompuServe Information Service (CIS). This includes applications such as CompuServe "navigators", but does not apply to general purpose GIF readers/writers (which are not intended for use primarily in conjunction with CIS).

On January 27, 1995, Unisys announced new licensing policies regarding "The Welch Patent". These include a .45% royalty on the total unit selling price of GIF/LZW products (minimum \$0.10, maximum \$10.00 per unit) and a .65% royalty on GIF/TIFF/LZW products (minimum \$0.20, maximum \$25.00). For further information and a copy of the written agreement it is possible to call Unisys at +1 215 986-4411, or send E-mail to <lzw\_info@unisys.com>.

Any organization using LZW should look at whether they have an infringement on Unisys' patent. CompuServe is not involved in any of these discussions - they are between Unisys and outside developers.

#### Software Patents

Normally, procedures such as LZW are published in magazines so that they can be shared by the community of software developers. LZW itself is a refinement of other algorithms published in the years before (Ziv-Lempel and others). Software is usually protected by copyright law, but in recent years (since 1981 in the USA) in several countries it has become possible to patent software. Initially, only software used to control hardware could be patented. This interpretation was soon extended to include all types of software (except for "pure mathematical algorithms"). While software patents have become an opportunity for many, they remain a controversial danger for others. Any programmer or publisher might be trapped at any time by a patent infringement claim that could not be foreseen or avoided.

Publication of an algorithm in a magazine does not automatically exclude a patent application. In many countries, including the USA, it is possible to apply for a patent and still publish the paper without mention of the application. In the USA (but not in many other countries), the patent application may even be filed within 12 months of the publication. Under such regulations, the only algorithms that might be used freely and without risk would be those published prior to 1981 (e.g. Donald Knuth's "The Art of Computer Programming").

Today, even designing a graphics file format can become a programmer's nightmare. One very active member of the Internet community (and author of the GZIP compressor) has collected information on more than 350 patents on lossless data compression and 100 on lossy image compression. Lempel, Ziv, Cohn and Eastman patented their original LZ78 algorithm (US patent 4,464,650). The LZW algorithm which is now attracting so much attention is patented by both IBM (4,814,746) and Unisys (4,558,302), while British Telecom (BT) holds a similar patent. The IBM patent application was filed

---

three weeks before that of Unisys, but the US patent office apparently failed to recognize that they covered the same algorithm. (The IBM patent is more general, but its claim 7 is said to be exactly LZW.)

#### 10 Years of LZW

While the original article on LZW was published in 1984, the LZW patent issue first surfaced in the press in 1989, when the BTLZ algorithm (a procedure similar to LZW developed and patented by British Telecom) was to be approved for data compression into the V.42bis modem standard. Unisys said on at least one occasion that it first began to learn of the widespread use of LZW in connection with the development of this standard. The first licensing arrangements put into place included those with modem manufacturers (\$ 20,000 for each one-time license) and with Adobe PostScript developers (\$ 10,000).

An article on "LZW Data Compression" was published in the October 1989 issue of Dr. Dobbs's Journal (see the Bibliography section for more details). A reader replied in the December issue explaining that the algorithm was patented. The author of the article added that he was unaware of any patent on the algorithm. More readers wrote, and in the March 1990 issue the editor-in-chief dedicated his Editorial to this topic, which in his words "sparked a forest of fires". The same issue also contained an official statement by Unisys Corporation, which confirmed that LZW was patented, mentioned the modem industry, and indicated how developers could contact Unisys.

In the October 2, 1989 issue of PC Week a columnist wrote:

"Alas, there's no consolation for developers of archiving programs that rely on the LZW data-compression algorithm. While cruising the bulletin boards last week, Spencer learned that Unisys has a patent on the algorithm, upon which a slew of data-compression programs are based. Watch out."

In about the same period, an article in InfoWorld mentioned the fact that modem manufacturers were facing the possibility of having to pay royalties to Unisys and to other patent holders for the right to use LZW.

Page 132 ("LZWEncode Filter") of the PostScript Language Reference Manual, Second Edition, published in December 1990, contains the address of the Welch Licensing Department at Unisys Corporation.

In the March 1991 issue of Byte, Steve Apiki ("Lossless Data Compression") explained that LZW is used in GIF, and that "The [LZW] algorithm itself is patented by Sperry [now Unisys]."

At this point, at least the readers of some publications were potentially aware of the LZW patent. But still, there were few links to GIF. Unisys apparently didn't know about GIF, nor did most GIF developers know that GIF contained LZW technology. And those who may have known, not necessarily knew about the patent.

This issue was also discussed among a small group of the better informed members of the CompuServe PICS Forum (now GRAPHSUP). The general feeling at that time was that "Unisys only intends to get royalties from hardware

vendors," and there was some consensus on the idea that Unisys "wouldn't do anything about pure software implementations".

Until the end of 1994, discussions on CompuServe's Information Service showed no clear mention of the requirement to get a license from Unisys for using LZW in GIF applications. During 1988 at least one developer stopped working on GIF tools because of considerations regarding the LZW patent, and reportedly "made CompuServe aware of it". This apparently was limited to private verbal conversations, and information on this behalf could be found neither in the press nor in CIS.

Among the developers who contacted Unisys between the end of 1990 and the beginning of 1991, there was at least one GIF developer. He recently described his experience:

"Finding the right person was the most difficult part of licensing LZW, but hopefully it's easier today (perhaps only 5 phone calls would be needed!)... When talking to Unisys back then, my recollection is that we had to basically tell the people at Unisys, 'Believe me, you DO own a patent on LZW; who do we talk to about LICENSING?' When we finally reached the licensing/legal department, THEY knew they had a patent, and spelled out the terms. I recall the person we were dealing with saying something like, 'They [Unisys] laugh when I make all these \$1 deals, but we have to charge something to protect the patent.'"

In those days, the standard license fee for PC-based software products was \$1 per copy sold (or a 1% royalty), after a \$100 advance payment. Apparently, Unisys still didn't know that GIF was based on LZW. In January 1995, Unisys stated: "Two years ago, Unisys learned that the LZW method was incorporated in the GIF specification and immediately began negotiations with CompuServe in January of 1993. We reached agreement with CompuServe on licensing the technology in June 1994..."

Two years before the Unisys statement, at the end of 1992, Cloanto, an Italian software house, contacted Unisys because it was interested in a license for the possible use of LZW in its PostScript Level 2 drivers. That correspondence also mentioned GIF and TIFF as using LZW, and anticipated some of the controversies which would follow 25 months later. Unisys replied: "... You raise a number of interesting issues which require consideration..."

While disclosing the full contents of this correspondence would probably not serve anyone's interest, the text of two letters sent to Unisys in 1992 is included at the end of this article, because the author feels that this 1992 perspective could complement the article with a few interesting ideas. The letters have not been edited, so some details (such as the reference to ZIP) may be incomplete with respect to current knowledge.

Unisys offered Cloanto a \$ .25 per unit royalty (1% of the net income) as an alternative to the PostScript one-time license, but did not answer the question raised by Cloanto: "If we implemented a software GIF or TIFF image file loader and saver (both formats are based on the LZW algorithm), would we need a license from Unisys Corp., as far as U.S. Patent 4,558,302 is concerned?". According to public statements, Unisys did however contact CompuServe the following month.

---

December 29, 1994 - The Days After

Between 1993 and 1994, the majority of developers still didn't know that GIF employed a patented algorithm, although both Unisys and CompuServe were aware of this (as the developers would learn in December 1994). Different opinions have been expressed on this. Some developers feel that reaching an agreement behind the scenes was the least destructive thing that could be done. Other (at times passionate) opinions picked up on electronic media are similar to these three:

"Consider this. CompuServe admits to knowing about patent problems with the GIF file format as early as January of 1993. ... We added GIF support to Fastgraph months after CompuServe admits knowledge of the patent problem... We relied on the information that was supplied to us by CompuServe. If CompuServe had told us the truth when they knew it, we never would have added GIF support..."

"If I chose to put GIF encode/decode functions in my software development toolkits, my main threat of legal liability would not come from Unisys, but rather from one of my customers being sued by Unisys, who would turn around and sue me for selling them some code that contained patented algorithms."

"I still don't have a clue what my situation is if I want to sell source and object code that imports and exports GIF images. I am not in the end-user app business, but my customers are, and they certainly will have to have an LZW license, but what about me? I've talked with Unisys by voice and E-mail, and the voice discussion was entirely unsatisfactory as I posted when it happened - basically the Unisys guy said anyone who sells code for \$100-\$300 a pop was a total \_\_\_\_\_ for selling it that cheap. The E-mail discussions I've had said 'OK - we hear you - we'll get back to you.' Never happened."

Unisys replied in part with reassuring clarifications to the general public, explaining that if the software was developed prior to 1995, or if it is public domain or freeware, the developer need not to worry:

"... Unisys does not intend to pursue previous inadvertent infringement by versions of GIF-based software products marketed prior to 1995... Unisys does not require licensing, or fees to be paid, for non-commercial, non-profit GIF-based applications, including those for use on the online services... Commercial developers... are expected to secure a licensing agreement with Unisys for software products introduced beginning in 1995, or enhancements of products that were introduced prior to 1995."

However, these statements were followed by far more restrictive interpretations. It soon became clear that Unisys could be demanding royalties for everything "manufactured" after 1994. One developer contacted Unisys and reported:

"I called the Unisys lawyer you referred me to and he confirmed this position. Even a book or CD containing \*pre 1995\* freeware is subject to royalties if the disk is put together in 1995... Royalties must be collected \*again\* for each update release."

While the new Unisys licensing policies (announced on January 27, 1995) enabled many software publishers to again ship their products after a month-long pause, other developers preferred to wait, hoping for a patent-free evolution of GIF. Comments included:

"What if I sign up and then they announce a new GIF specification which does not use LZW?"

"Labeling and user notification requirements in the agreement are ridiculous. I understand their desire to 'spread the word' about their patent, but they're telling me that I have to provide far more info on their ownership of the patent than they require in the docs/packaging of modem manufacturers and other users of LZW. Fair is fair. A blurb in the online help and docs should be sufficient; a 'non-defeatable' splash screen at startup is going too far."

"Unisys is attempting to control how we (and other shareware authors) do business, and to make us billboards for their LZW patent... By making me tell my users how many security backups they can make, etc., they're telling me how to run my business and how to interface with my customers."

"Imagine the nightmare of having to pay royalties to 10 patent holders, each of whom tells you how to run your business..."

"Unisys has given us a chance to work together to change the system - rather than waiting to be sued one by one for this patent or that. We can win the fight against software patents, if we speak loud and clear against them."

Some of the most active developers decided to collaborate on the design of a patent-free evolution of GIF (and TIFF's LZW compression mode). A variety of different procedures and data structures (such as Shannon-Fano and AVL trees) have been used to compress data in ways similar, if not equivalent, to LZW. But this diversity apparently does not escape the patent. As one expert said, "If the output data is GIF, the compressor infringes the Unisys patent regardless of the algorithm."

On January 16, 1995, CompuServe declared its intention to coordinate the development of GIF24, a freely usable successor to GIF capable of 24-bit lossless compression. Several developers invested a lot of time and energies to solve the Unisys patent problem, and rapidly worked out different modifications to the GIF specification. One of the better known efforts was the project for a "GEF" graphics-exchange format. GEF and GIF24 converged into PNG (official abbreviation of "Portable Network Graphics", unofficially "Png is Not Gif").

The open architecture of PNG preserves the simplicity that made GIF so popular, and adds features such as true color. Test results indicate that PNG is capable of (losslessly) compressing true color images better than any other widely used image format. It is also more effective than GIF in storing palette-based images. (More information on PNG is included in the Reference and Bibliography sections.)

At the end, it appears that if so many efforts converge into a new,

---



improved standard, we still have to give part of the credit to the LZW patent...

The author of this text can be contacted at <mcb@cloanto.it>.  
Any comments, or experience you would like to share, would  
be very appreciated.

#### Reference

If the excerpts from the PNG specification are not included here in order to keep the file size reasonable ("lossy compression"), please check for another file accompanying this text (suggested file name: "giflzw2.txt"), or send E-mail to <gltext2@cloanto.it> before June 30, 1995. The latest hypertext version of the full document is available on the World Wide Web: <URL:http://sunsite.unc.edu/boutell/png.html>

---

Excerpts from the PNG (Portable Network Graphics) Specification,  
Ninth Draft - Revision date: 7 March, 1995

[The text is not included here - Newer versions are available]

---

If the official texts from CompuServe and Unisys are not included here in order to keep the file size reasonable, please check for another file accompanying this text (suggested file name: "giflzw2.txt"), or send E-mail to <gltext2@cloanto.it> before June 30, 1995.

---

#### AGREEMENT FOR USE OF GRAPHICS INTERCHANGE FORMAT (SM)

[The text of the Graphics Interchange Format (GIF) Developer Agreement, released by CompuServe on December 29, 1994 is not included here. It became obsolete when Unisys announced its new licensing policies regarding "The Welch Patent" on January 27, 1995.]

---

---

%: 174559 S19/GIF/LZW Discussion [GRAPHSUP]  
04-Jan-95 17:07:50  
Sb: #GIF/LZW Clarification  
Fm: Larry Wood 76703,704  
To: All

[The text is not included here]

---

---

#: 181065 S3/Hot News and Rumor  
07-Jan-95 19:12:19  
Sb: #Unisys GIF Clarification  
Fm: Steve Ahlstrom/SYSOP 76703,2006  
To: All

[The text is not included here]

---

---

From: rmarks@ecdcsvr.tredydev.unisys.com (Richard Marks)  
Date: Fri, 6 Jan 1995 22:09:14 GMT

Unisys LZW Patent FREQUENTLY ASKED QUESTIONS January 6, 1995

[The text is not included here]

---

If the texts of the two letters are not included here in order to keep the file size reasonable, please check for another file accompanying this text (suggested file name: "giflzw2.txt"), or send E-mail to <gltext2@cloanto.it> before June 30, 1995.

---

From Cloanto/Unisys - November 6, 1992

[The text is not included here]

---

---

From Cloanto/Unisys - November 12, 1992

[The text is not included here]

---

#### Bibliography

[Author Unknown - Information Appreciated]  
"Spencer the Katt"  
PC Week, October 2, 1989

Adobe Systems Incorporated  
"LZWEncode Filter"  
PostScript Language Reference Manual, Second Edition  
Addison-Wesley Publishing Company

---

ISBN 0-201-18127-4

Apiki, Steve

"Lossless Data Compression"

Byte, March 1991, pages 309-314, 386-387

Association of Shareware Professionals Forum

CompuServe GO ASPFORUM

Bell, Timothy C., Cleary, John G. and Witten, Ian H.

"Adaptive Dictionary Encoders"

Text Compression

Prentice Hall

ISBN 0-13-911991-4

Boutell, Thomas (Editor)

PNG (Portable Network Graphics) Specification

Ninth Draft - Revision date: 7 March, 1995

Hypertext version available on the World Wide Web:

<URL:<http://sunsite.unc.edu/boutell/png.html>>

Clay, Betty

"Texas Tales"

ICPUG Newsletter, January/February 1995, pages 18-23

Cloanto Italia srl

Supplement to Personal Paint Manual

Version 6.1/1995, January 27, 1995

CompuServe Graphics Developers Forum (GO GRAPHDEV)

CompuServe Graphics Support Forum (GO GRAPHSUP)

Console Battilana, Michele

"LZW Data Compression without Hashing"

University of Udine Exam Project, July 9, 1987

Elmer-Dewitt, Philip

"Will Gates Get the Net?"

Time, January 30, 1995, Page 47

Erickson, Jonathan

"Patent Letter Suits" (Editorial)

Dr. Dobb's Journal, March 1990, page 6

Erickson, Jonathan

"The Green, Green Cash of Gnomes" (Editorial)

Dr. Dobb's Journal, April 1995, page 6

Gardner, Ray

"LZW Patent Issues" (Letter)

Dr. Dobb's Journal, December 1989, page 8

Internet comp.graphics Newsgroups

Internet comp.sys.graphics Newsgroup

---

Knuth, Donald E.  
The Art of Computer Programming  
Volume 3 / Sorting and Searching  
Addison-Wesley Publishing Company  
ISBN 0-201-03803-X

Landy, Gene K.  
The Software Developer's and Marketer's Legal Companion  
Addison-Wesley Publishing Company  
ISBN 0-201-62276-9

Miles, J. B.  
"Patent Issues May Stall Approval of New V.42bis Modem Standard"  
InfoWorld, approximately fall of 1989, pages 43-44  
[Author, Article Title and Exact Date Unknown - Information Appreciated]  
[InfoWorld Article on LZW and Modem Implementations - Is this it?]

Nelson, Mark R.  
"LZW Data Compression"  
Dr. Dobbs's Journal, October 1989, pages 29-36, 86-87

Nelson, Mark R.  
"LZW Patent Issues" (Reply to Letter)  
Dr. Dobbs's Journal, December 1989, pages 8-12

PNG (Portable Network Graphics)  
Information and support material available from:  
Internet comp.graphics Newsgroups  
Internet comp.sys.graphics Newsgroup  
CompuServe Graphics Support Forum (GO GRAPHSUP)  
Via FTP from <ftp://godzilli.cs.sunysb.edu/pub/ngf>  
The PNG specification is also available on the World Wide Web:  
<URL:http://sunsite.unc.edu/boutell/png.html>

Unisys Corporation  
"Patented Algorithms" (Letter)  
Dr. Dobbs's Journal, March 1990, page 8

Vaughan-Nichols, Steven J.  
"Saving Space", "Squeeze, Squash, and Crush" and "Legal Seagull"  
Byte, March 1990, pages 237-243

Welch, Terry A.  
"A technique for high-performance data compression"  
IEEE Computer, June 1984, pages 8-19

Ziv, Jacob and Lempel, Abraham  
"A universal algorithm for sequential data compression"  
IEEE Transactions on Information Theory, May 1977, pages 337-343

Ziv, Jacob and Lempel, Abraham  
"Compression of individual sequences via variable-rate coding"  
IEEE Transactions on Information Theory, September 1978, pages 530-536

Special thanks to Dave, David, Diana, Frank, Jason, Jean-loup, Jon, Kevin,

---

Larry, Pierce, Richard, Tim, Tom and many others for their precious help.

## 1.6 PNG Specification

PNG (Portable Network Graphics) Specification

Information and support material is available from:

```
Internet comp.graphics Newsgroups
Internet comp.sys.graphics Newsgroup
CompuServe Graphics Support Forum (GO GRAPHSUP)
Via FTP from <ftp://godzilli.cs.sunysb.edu/pub/ngf>
```

The PNG specification is also available on the World Wide Web:

```
<URL:http://sunsite.unc.edu/boutell/png.html>
```

## 1.7 GIF to PNG ARexx Script

```
/* GifToPng.pprx - PPaint ARexx Script - Copyright © 1995 Cloanto Italia srl
```

```
$VER: GifToPng.pprx 1.0
```

```
More Comments
```

```
*/
```

```
MYPORT = 'PPAINT'
```

```
IF ~SHOW('P', MYPORT) THEN DO
```

```
  IF EXISTS('PPaint:PPaint') THEN DO
```

```
    ADDRESS COMMAND 'Run >NIL: PPaint:PPaint'
```

```
    DO 30 WHILE ~SHOW('P', MYPORT)
```

```
      ADDRESS COMMAND 'Wait >NIL: 1 SEC'
```

```
    END
```

```
  END
```

```
ELSE DO
```

```
  SAY "Personal Paint could not be loaded."
```

```
  EXIT 10
```

```
END
```

```
END
```

```
IF ~SHOW('P', MYPORT) THEN DO
```

```
  SAY 'Personal Paint REXX port could not be opened.'
```

```
  EXIT 10
```

```
END
```

```
ADDRESS VALUE MYPORT
```

```
OPTIONS RESULTS
```

```
OPTIONS FAILAT 10000
```

```
Version 'REXX'
```

```
IF RESULT < 2 THEN DO
```

```

    errmess = 'This script requires a newer_version of Personal Paint.'
    SAY errmess
    RequestNotify 'PROMPT="'errmess'"
    EXIT 10
END

LockGUI
FreeBrush
IF RC = 0 THEN RequestPath 'GifToPng target directory'
IF RC = 0 THEN DO
    tmpfname = 'T:pprx_temp.'PRAGMA('ID')
    ADDRESS COMMAND 'List >'tmpfname' 'RESULT' NOHEAD PAT=~(#?.info) LFORMAT="*"s%s ←
        *"" ALL FILES'
    IF OPEN('listfile', tmpfname, R) THEN DO
        Get SETTING ICONS
        iconmode = RESULT
        errcode = 0
        Set '"ICONS=3"'
        DO FOREVER
            curfname = READLN('listfile')
            IF EOF('listfile') THEN BREAK
            GetFileFormat curfname
            IF RC = 0 THEN DO
                IF UPPER(RESULT) = 'GIF' THEN DO
                    SAY 'Processing 'curfname
                    LoadBrush curfname FORCE
                    IF RC = 0 THEN DO
                        IF UPPER(RIGHT(curfname, 5)) = '.GIF"' THEN DO
                            len = LENGTH(curfname)
                            newfname = OVERLAY(D2C(C2D(SUBSTR(curfname, len-3, 1)) + 9), ←
                                curfname, len-3)
                            newfname = OVERLAY(D2C(C2D(SUBSTR(curfname, len-2, 1)) + 5), ←
                                newfname, len-2)
                            newfname = OVERLAY(D2C(C2D(SUBSTR(curfname, len-1, 1)) + 1), ←
                                newfname, len-1)
                            IF EXISTS(SUBSTR(newfname, 2, len-2)) = 0 THEN DO
                                ADDRESS COMMAND 'Rename >NIL: 'curfname' 'newfname
                                curiconfname = INSERT('.info', curfname, len-1)
                                newiconfname = INSERT('.info', newfname, len-1)
                                curfname = newfname
                                IF EXISTS(SUBSTR(curiconfname, 2, len+3)) THEN DO
                                    IF EXISTS(SUBSTR(newiconfname, 2, len+3)) THEN
                                        ADDRESS COMMAND 'Delete >NIL: 'curiconfname
                                    ELSE
                                        ADDRESS COMMAND 'Rename >NIL: 'curiconfname' 'newiconfname
                                    END
                                END
                            END
                        END
                    END
                END
            END
            SaveBrush 'FORCE FILE 'curfname' FORMAT PNG OPTIONS "PROGDSP=0" "COMPR ←
                =6" "AUTO=1"'
            IF RC > 0 THEN DO
                errcode = RC
                IF RC = 5 THEN
                    errmess = 'User abort during save.'
                ELSE
                    errmess = 'Error 'RC' during save.'
                END
            END
        END
    END

```

---

```

        FreeBrush FORCE
    END
    ELSE DO
        errcode = RC
        IF RC = 5 THEN
            errmess = 'User abort during load.'
        ELSE
            errmess = 'Error 'RC' during load.'
        END
    END
END
END
IF errcode > 0 THEN BREAK
END
IF errcode > 0 THEN DO
    SAY errmess
    RequestNotify 'PROMPT="'errmess'"'
END
Set ' "ICONS='iconmode'"
CALL CLOSE('listfile')
END
ADDRESS COMMAND 'Delete >NIL: 'tmpfname
END
UnlockGUI

```

## 1.8 GIF to PNG ARexx Script - Comments

### GIF to PNG ARexx Script - Comments

This script asks the user to specify a directory, scans the directory and its subdirectories and converts all GIF files it finds into PNG.

Non-GIF files are not affected. Icon images are preserved. Icon format information is updated (Tool Types: FILETYPE=PNG). GIF Author, Copyright and Comment fields are translated to PNG equivalents. File name suffixes are changed (i.e. the files are renamed) as follows:

```

.gif    -> .png
.GIF    -> .PNG
.Gif    -> .Png, ecc.

```

others -> unchanged

Personal Paint identifies the file type by its contents (not by the file name suffix). If the script runs during Workbench use, the Workbench Update menu item must be selected to visually update the contents of any windows containing files being renamed by this script.

This script requires Personal Paint version 6.3 (PPaint Rexx version 2) or higher, `personal_png_io.library` (enclosed with PPaint), and `personal_gif_io.library` (available via free electronic distribution).

Possible changes that could be applied to this file:

Convert all images to PNG (not just GIFs). To do this, change the line

selecting GIFs to IF UPPER(RESULT) ~= 'PNG' THEN DO. However, be careful if you have IFF animations, as they can be loaded as ILBM images unless they are filtered out (IFF animations begin with an ILBM image).

Activate PNG Adam 7 progressive display in files being written. This degrades compression but the resulting images appear more nicely when displayed by progressive viewers. Set PROGDSP=1.

Convert any file to uncompressed IFF-ILBM. This may be good for files to be stored on an Amiga CD-ROM, where loading speed could be more important than compression. Remove the instructions selecting only GIFs and replace the PNG FORMAT option with FORMAT ILBM OPTIONS "COMPR=0" "SCRFMT=0".

PNG was designed as a replacement and extension to GIF and LZW-based TIFF, after Unisys Corporation began demanding royalties on GIF/LZW code. As the PNG specification was released in May 1995, it gained general recognition as the best lossless standard for storing digital images.

Cloanto, the first software house to publish a paint program supporting the PNG file format, is also making available a PNG developer's kit for the Amiga. This includes instructions for using `personal_png_io.library`, and is a commercial product. A PNG DataType is available at no cost for free electronic distribution.

An Article on the GIF/LZW issue can be requested by sending E-mail to

`<gltext1@cloanto.it>`

For more information, or suggestions, please address E-mail to

`<info@cloanto.it>`

## 1.9 DataTypes

### Amiga DataTypes

Amiga DataTypes provide an object oriented approach for determining and handling data types. A simple DataType consists of one short format descriptor file (stored in `DEVS:DataTypes`) and a library containing code to process that datatype (stored in `SYS:Classes/DataTypes`).

A format descriptor file simply describes how to recognize a file. For example, it could list the initial bytes (if all files of that format have a standard header). This is how the PNG format descriptor works. The system of format descriptions can be used by applications to identify various file formats (for example, to assign appropriate Workbench icons to files that don't come with an icon). Beyond that, the DataType I/O libraries allow applications to handle different formats without re-implementing the same format again and again.

In theory, DataTypes should be documented in the original Amiga documentation. In the 3.1 Amiga Developer Update disk set, documentation can be found in `Docs/datatypes.doc`, `Tutorial/DataTypes` and `Examples2/DataTypes`.

---



In practice, the original release of the documentation was not very complete, and it has allowed very few developers to implement support for DataTypes in their applications. At Cloanto we felt the same problem, and for this reason we would now like to share some of our own source code with other developers.

The C source code of a DataType-based image viewer is included here.

## 1.10 C Source Code

```

/*
  ViewDT

  Simple DataType-based Picture Viewer

  Syntax:
    ViewDT FILES/A

  Examples:
    ViewDT pictures:space/#?.pic
    ViewDT portrait.png

  Status:
    Public Domain

  If you require more information please send E-mail to <info@cloanto.it>
*/

#include <exec/types.h>
#include <exec/memory.h>
#include <graphics/gfx.h>
#include <graphics/displayinfo.h>
#include <intuition/intuitionbase.h>
#include <intuition/gadgetclass.h>
#include <datatypes/datatypes.h>
#include <datatypes/datatypesclass.h>
#include <datatypes/pictureclass.h>
#include <proto/exec.h>
#include <proto/dos.h>
#include <proto/graphics.h>
#include <proto/intuition.h>
#include <proto/datatypes.h>
#include <string.h>
#include <stdio.h>

struct Picture
{
    struct BitMapHeader bmhd; /* format and infos */
    struct BitMap *bmap;     /* bitmap */
    ULONG *palette;          /* color table in LoadRGB32() format */
    LONG palette_size;        /* mem usage */
    LONG palette_entries;     /* number of colors */
    ULONG display_ID;         /* video mode */
    UBYTE *author;           /* author info */

```

---

```

    UBYTE *copyright;          /* copyright info */
    UBYTE *annotation;        /* other info */
    LONG author_size;          /* mem usage */
    LONG copyright_size;        /* mem usage */
    LONG annotation_size;       /* mem usage */
};

void FreePicture(struct Picture *pic);
LONG GetDataTypesPicture(UBYTE *file_name, struct Picture *pic, ULONG);
BOOL IsDataTypes(UBYTE *file_name, UBYTE *name_buff, LONG nbuff_size);
BOOL ViewPicture(struct Picture *pic);

struct IntuitionBase *IntuitionBase;
struct GfxBase *GfxBase;
struct Library *DataTypesBase;

/*
    IsDataTypes

    Parameters
        file_name: name of the file to inspect
        name_buff: (optional) buffer to store the file format name
        nbuff_size: size of name_buff

    Return value
        TRUE if DataTypes recognized the file as a valid picture file
        FALSE otherwise
*/
BOOL IsDataTypes(UBYTE *file_name, UBYTE *name_buff, LONG nbuff_size)
{
    struct DataType *dtn;
    struct DataTypeHeader *dth;
    BPTR lock;
    BOOL it_is;

    it_is = FALSE;
    if (lock = Lock(file_name, ACCESS_READ))
    {
        /* inspect file */
        if (dtn = ObtainDataTypeA(DTST_FILE, (APTR)lock, NULL))
        {
            dth = dtn->dtn_Header;
            if (dth->dth_GroupID == GID_PICTURE)    /* is it a picture? */
            {
                it_is = TRUE;
                if (name_buff)
                {
                    strncpy(name_buff, dth->dth_Name, nbuff_size);
                    *(name_buff + nbuff_size - 1) = 0; /* safe strncpy() termination ↵
                */
            }
        }
        ReleaseDataType(dtn);
    }
    UnLock(lock);
}

```

---

```

    }
    return(it_is);
}

/*
GetDataTypesPicture

Parameters
    file_name: name of the file to load
    pic: work structure
    bmap_flags: AllocBitMap() flags (e.g. BMF_DISPLAYABLE, BMF_INTERLEAVED etc.)

Return value
    0 if successful (picture info and data in "pic" structure) or
    error code as from dos.library IoErr() function
*/
LONG GetDataTypesPicture(UBYTE *file_name, struct Picture *pic, ULONG bmap_flags)
{
    Object *obj;
    struct BitMapHeader *bmh;
    struct BitMap *bmap;
    struct gpLayout layout;
    ULONG *creg, *ctab;
    UBYTE *str;
    LONG ncol, crsize, err;

    memset(pic, 0, sizeof(struct Picture)); /* clear pic structure */
    err = 0;

    if (obj = NewDTObject(file_name,
                          DTA_SourceType, DTST_FILE,
                          DTA_GroupID, GID_PICTURE,
                          PDTA_Remap, FALSE,
                          TAG_DONE)) /* get the picture object */
    {
        if (GetDTAttrs(obj,
                       PDTA_ModeID, &pic->display_ID,
                       PDTA_BitMapHeader, &bmh,
                       TAG_DONE) == 2) /* get the bitmap_header and mode_id */
        {
            pic->bmhd = *bmh;

            /*
            query the object about its author, copyright and annotation
            */
            if (GetDTAttrs(obj, DTA_ObjAuthor, &str, TAG_DONE) == 1)
            {
                if (str)
                {
                    pic->author_size = strlen(str) + 1;
                    if (pic->author = AllocMem(pic->author_size, 0))
                        strcpy(pic->author, str);
                }
            }
            if (GetDTAttrs(obj, DTA_ObjCopyright, &str, TAG_DONE) == 1)
            {
                if (str)

```

---

```

    {
        pic->copyright_size = strlen(str) + 1;
        if (pic->copyright = AllocMem(pic->copyright_size, 0))
            strcpy(pic->copyright, str);
    }
}
if (GetDTAttrs(obj, DTA_ObjAnnotation, &str, TAG_DONE) == 1)
{
    if (str)
    {
        pic->annotation_size = strlen(str) + 1;
        if (pic->annotation = AllocMem(pic->annotation_size, 0))
            strcpy(pic->annotation, str);
    }
}

layout.MethodID = DTM_PROCLAYOUT;    /* render the object */
layout.gpl_GInfo = NULL;
layout.gpl_Initial = TRUE;

if (DoDTMethodA(obj, NULL, NULL, (Msg)&layout))
{
    if (GetDTAttrs(obj,
                    PDTA_BitMap, &bmap,
                    PDTA_CRegs, &creg,
                    PDTA_NumColors, &ncol,
                    TAG_DONE) == 3)    /* get the bitmap and its colors */
    {
        crsize = (ncol * 3) * 4;
        pic->palette_entries = ncol;
        pic->palette_size = crsize + (2 * 4);    /* LoadRGB32() table ←
            requirements */

        if (pic->palette = AllocMem(pic->palette_size, 0))
        {
            ctab = pic->palette;
            *ctab++ = (ncol << 16) | 0;    // number of colors and first ←
                color to load
            memcpy(ctab, creg, crsize);
            *(ctab + (crsize / 4)) = 0;    // terminator
        }
        else err = ERROR_NO_FREE_STORE;

        if (pic->bmap = AllocBitMap(pic->bmhd.bmh_Width, pic->bmhd. ←
            bmh_Height, pic->bmhd.bmh_Depth, bmap_flags, NULL))
        {
            BltBitMap(bmap, 0,0, pic->bmap, 0,0, pic->bmhd.bmh_Width, pic-> ←
                bmhd.bmh_Height, 0xC0, 0xFF, NULL);
            WaitBlit();
        }
        else err = ERROR_NO_FREE_STORE;
    }
    else err = IoErr();
}
else err = IoErr();
}
else err = ERROR_REQUIRED_ARG_MISSING;

```

---

```
    DisposeDTObject(obj);    /* free the object */
}
else err = IoErr();

if (err)
    FreePicture(pic);

return(err);
}

/*
FreePicture

Parameters
    pic: Picture structure with resources to free

Return value
    none
*/
void FreePicture(struct Picture *pic)
{
    if (pic->bmap)
    {
        WaitBlit();
        FreeBitMap(pic->bmap);
    }
    if (pic->palette)
        FreeMem(pic->palette, pic->palette_size);

    if (pic->author)
        FreeMem(pic->author, pic->author_size);

    if (pic->copyright)
        FreeMem(pic->copyright, pic->copyright_size);

    if (pic->annotation)
        FreeMem(pic->annotation, pic->annotation_size);

    memset(pic, 0, sizeof(struct Picture));    /* clear it all */
}

/*
ViewPicture

Parameters
    pic: picture infos and data

Return value
    TRUE if the user cancelled the view sequence (<Esc> key)
    FALSE otherwise
*/
BOOL ViewPicture(struct Picture *pic)
{
    struct Screen *scr;
    struct Window *win;
    struct IntuiMessage *imsg;
```

---

```

BOOL done, quit;

done = quit = FALSE;
if (scr = OpenScreenTags(NULL,
    SA_Width, pic->bmhd.bmh_Width,
    SA_Height, pic->bmhd.bmh_Height,
    SA_Depth, pic->bmhd.bmh_Depth,
    SA_Quiet, TRUE,
    SA_ShowTitle, FALSE, /* no title bar */
    SA_Behind, TRUE,
    SA_Type, CUSTOMSCREEN,
    SA_DisplayID, pic->display_ID,
    SA_Overscan, OSCAN_TEXT,
    SA_AutoScroll, TRUE,
    SA_Colors32, pic->palette,
    SA_BackFill, LAYERS_NOBACKFILL, /* no screen-clearing when the ↵
        window is closed (is faster) */
    TAG_END))
{
    if (win = OpenWindowTags(NULL,
        WA_Width, scr->Width,
        WA_Height, scr->Height,
        WA_IDCMP, MOUSEBUTTONS | VANILLAKEY,
        WA_CustomScreen, scr,
        WA_Backdrop, TRUE,
        WA_Borderless, TRUE,
        WA_Activate, TRUE,
        WA_RMBTrap, TRUE,
        WA_SimpleRefresh, TRUE,
        WA_BackFill, LAYERS_NOBACKFILL, /* no screen-clearing when the ↵
            window is opened (is faster) */
        TAG_END))
    {
        BltBitMap(pic->bmap, 0,0, scr->RastPort.BitMap, 0,0, pic->bmhd.bmh_Width, ↵
            pic->bmhd.bmh_Height, 0xC0, 0xFF, NULL);
        WaitBlit();
        ScreenToFront(scr); /* show the screen only when the picture has been ↵
            copied to it */

        while (!done)
        {
            Wait(1 << win->UserPort->mp_SigBit);

            while (imsg = (struct IntuiMessage *)GetMsg(win->UserPort))
            {
                switch (imsg->Class)
                {
                    case VANILLAKEY:
                        switch (imsg->Code)
                        {
                            case 27: done = quit = TRUE; break; /* <Esc> = cancel ↵
                                */
                            case 13:
                            case 32: done = TRUE; break; /* <Enter> / <Space> = ↵
                                continue */
                        }
                        break;
                }
            }
        }
    }
}

```

---

```

        case MOUSEBUTTONS:
            switch (imsg->Code)
            {
                case SELECTUP:      /* MButton = continue */
                case MIDDLEUP:
                case MENUUP:        done = TRUE; break;
            }
            break;
        }
        ReplyMsg((struct Message *)imsg);
    }
}
CloseWindow(win);
}
CloseScreen(scr);
}
return(quit);
}

void main(int argc, char *argv[])
{
    #define AP_BUFFSIZE  240
    #define PT_SIZE      80
    struct Picture pic;
    struct AnchorPath *ap;
    UBYTE pic_type[PT_SIZE];
    LONG err;
    BOOL quit;

    if (argc > 1)
    {
        if (IntuitionBase = (struct IntuitionBase *)OpenLibrary("intuition.library", ↵
            39))
        {
            if (GfxBase = (struct GfxBase *)OpenLibrary("graphics.library", 39))
            {
                if (DataTypesBase = OpenLibrary("datatypes.library", 39))
                {
                    if (ap = AllocMem(sizeof(struct AnchorPath) + AP_BUFFSIZE, ↵
                        MEMF_CLEAR))
                    {
                        ap->ap_Strlen = AP_BUFFSIZE;
                        for (err = MatchFirst(argv[1], ap); err == 0; err = MatchNext(ap ↵
                            ))
                        {
                            if (IsDataTypes(ap->ap_Buf, pic_type, PT_SIZE))
                            {
                                if (GetDataTypesPicture(ap->ap_Buf, &pic, 0) == 0)
                                {
                                    printf("%s (%s, %dx%d, %d colors)\n",
                                        ap->ap_Info.fib_FileName,
                                        pic_type,
                                        pic.bmhd.bmh_Width,
                                        pic.bmhd.bmh_Height,
                                        pic.palette_entries);
                                    if (pic.author)
                                        printf("    author: %s\n", pic.author);
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```

---

```

        if (pic.copyright)
            printf("    copyright: %s\n", pic.copyright);
        if (pic.annotation)
            printf("    annotation: %s\n", pic.annotation);

        quit = ViewPicture(&pic);
        FreePicture(&pic);
        if (quit)
            break;
    }
}
}
MatchEnd(ap);
FreeMem(ap, sizeof(struct AnchorPath) + AP_BUFFSIZE);
}
CloseLibrary(DataTypesBase);
}
CloseLibrary((struct Library *)GfxBase);
}
CloseLibrary((struct Library *)IntuitionBase);
}
}
}

```

## 1.11 Personal Paint

### Personal Paint 6.3

Personal Paint is a powerful and intuitive paint, image processing, animation and 24-bit printing package. Employ stunning effects like transparencies, emboss, water-colors and stereograms (as in "Magic Eye"), while virtual memory frees precious Chip RAM by using other storage resources! Plus: support of RTG graphics cards, different file formats (IFF, PNG, PCX, encrypted, C source code, DataTypes etc.), nine brushes, two independent working environments, multi-level Undo/Redo, animation storyboard, Bézier curves, autoscroll painting, professional color reduction, superior text editor, color fonts, PostScript output, screen grabber, ARexx...

Features of Personal Paint 6.3 Include:

- First paint program worldwide to support the new PNG (Portable Network Graphics) file format. Includes an ARexx script to convert GIFs to PNG.
- Animation (featuring a storyboard, superior compression, multiple palettes, frame-by-frame timing, ANIM-5/7/8 and hybrid formats, etc.)
- Sophisticated "behind the scenes" memory management, including virtual memory (swaps inactive image data to Fast RAM and disk storage) and multiple levels of undo and redo
- New, faster image processing effects, including transparencies, alpha channel and single image stereograms (both SIRDs and custom pattern stereograms, as in "Magic Eye")



- Support of Retargetable Graphics (display cards like the Picasso, Retina, Piccolo, Rainbow, EGS, Talon, Cybergraphics etc.)
- Animation on RTG display cards (with or without double-buffering)
- Direct, high quality 24-bit printing (Color and Black & White) and interface to third-party software such as Studio Print Server
- Professional and fast modes for converting 24-bit pictures (IFF, PNG, PCX, PBM etc.) to 256 colors or less
- HAM, HAM8 and Picasso 24-bit viewer active during color reduction
- External input/output modules (loaders and savers) for easy extensions and upgrades. Modules for IFF, PNG, PCX, PBM, Amiga DataTypes and several others are included. GIF module is available from public domain sources.
- Support and editing of IFF, PNG and GIF project annotations (Author, Copyright and Comment fields, plus Amiga filenotes)
- Autoscroll painting
- Workbench Application Icon (Drag and Drop)
- Basic set of ARexx commands for presentations, format conversions and printing
- ASL-compatible file requester
- More power through machine language code: the software is in part up to 500% faster
- "New Look" user interface
- A collection of utilities, including color fonts, new DeskJet printer drivers (up to four inks) and JPEG DataType

#### Requirements:

- 1 Mbyte RAM, 1 disk drive required; 1 Mbyte of Chip RAM recommended
- Amiga Kickstart from 1.2, Amiga Workbench from 1.3
- Actively exploits 2.x and 3.x operating systems, FPU, CPU Cache RAM and RTG boards
- JPEG DataType requires at least 68020 CPU

#### For more information:

Distributors

Orders

---

Cloanto

## 1.12 Orders

Orders

For availability and pricing information regarding all Cloanto packages and upgrades, please contact Cloanto. Orders can be e-mailed to <orders@cloanto.it>.

If you wish to order with credit card payment, please indicate the card number, expiration date and the cardholder's name and date of birth. If no price is indicated in the order, it will be indicated to you for confirmation before proceeding.

For more information:

Distributors

Cloanto

## 1.13 Distributors

Cloanto Distributors

North America:

DKB

29318 Lorie Lane

USA - Wixom, MI 48393

Tel +1 810 348-3755 (tech.)

Tel +1 810 348-3821 (sales)

Fax +1 810 348-3531

Great Britain:

Ramiga International Ltd.

Stablau 'Rin

GB - Pentrefoelas, Clwyd LL24 0HT

Tel +44 1690 770304

Fax +44 1690 770266

Bbs +44 1690 770696

Meridian Software Distribution Ltd.

East House

East Road Industrial Estate

GB - London SW19 1AH

Tel +44 181 5433500

Fax +44 181 5432255

German Area:

Casablanca Multimedia GmbH

Wiemelhauser Straße 247 a

D - 44799 Bochum

Tel +49 234 72035

Fax +49 234 72060

The Netherlands:

Barlage Computer Hardware

Kaalheidersteenweg 262

NL - 6467 AH Kerkrade

Tel +31 45 425881

Fax +31 45 424411

Danemark:  
Epic Data Tel +45 59 93 10 25  
Østerlyngvej 1 Fax +45 59 93 20 54  
DK - 4500 Nykøbing Sjælland

Sweden:  
ProComp Tel +46 472 70845  
Box 46 Fax +46 472 71680  
S - 340 36 Moheda Bbs +46 472 71270

France:  
Cuda Informatique sarl Tel +33 1 42 46 47 48  
31, rue de Trevis Fax +33 1 42 46 47 01  
F - 75009 Paris

Greece:  
Multi-RAK Tel +49 234 9489411  
Karnassiotis & Radouniklis GBR Fax +49 234 9489444  
Prinz-Regent-Straße 70  
D - 44795 Bochum

Australia and New Zealand:  
Amadeus Computers Tel +61 2 652-2712  
34 Tecoma Drive Fax +61 2 652-1515  
Glenorie NSW 2157  
Australia

Italy:  
Db-Line srl Tel +39 332 767270  
Viale Rimembranze 26/C Fax +39 332 767244  
I - 21024 Biandronno VA Bbs +39 332 767329

Italy (OEM):  
CTO spa Tel +39 51 753133  
Via Piemonte 7/f Fax +39 51 753418  
I - 40069 Zola Predosa BO

For more information:

Orders

Cloanto

## 1.14 Legal Notes

### Legal Notes

These notes apply to the PNG DataType and support files, stored in the PNG\_dt.lha archive. The material is offered at no charge for personal use.

Copyright

---

The collection and the individual files stored in this archive are Copyright © 1995 Cloanto Italia srl (Cloanto), unless otherwise indicated in the files or in the following sentence.

The PNG format descriptor file (92 bytes of length, stored in DEVS:DataTypes) and the C source code example (ViewDT.c) contained in this document are hereby placed in the public domain.

#### Disclaimer and Limitation of Liability

This software is provided "as is", without warranty of any kind, either expressed or implied, statutory or otherwise. By using the archive and its contents, you accept the entire risk as to its quality and performance.

Neither Cloanto nor any other party involved in the creation, production or delivery of the archive and its contents shall be liable for any direct, indirect, special, consequential or incidental damages, including without limitation damages for loss of profits, loss of use or loss of anticipated costs, expenses or damages, and any data or information which may be lost or rendered inaccurate, even if Cloanto is advised of the possibility of such damages.

#### Distribution

Cloanto authorizes all types of online (electronic) distribution of the unmodified archive, as long as the archive is neither sold nor included in another product. Other types of distribution, and storage on magnetic, optical or other type of media other than for online distribution and personal use require written permission or license from Cloanto. The following section ("Specific Publications") contains a permission valid for certain publications.

#### Specific Publications

Floppy disk based "public domain" collections and distributors thereof charging only a nominal fee are hereby authorized to include the unmodified archive. This permission does not include CD-ROM collections and magazine cover disks, nor does it grant any rights to use the "Cloanto" trademark.

Cloanto hereby specifically authorizes the inclusion of this archive on Fred Fish compilations by Amiga Library Services, Aminet by Urban D. Müller and Meeting Pearls by Angela Schmidt. These organizations may also include the complete files of this archive in uncompressed form (please make sure that the original file dates are preserved), on floppy disk and/or CD-ROM. This permission does not include other organizations copying or republishing these collections.

If you plan to make the archive available to others, please consider contacting Cloanto (E-mail <info@cloanto.it>) to check if a newer version exists. Sample copies are always appreciated, but not required.

#### Trademarks

---

Cloanto and the Cloanto logo are registered trademarks, and Personal Fonts Maker, PFM, Personal Paint, PPaint, Personal Write and PWrite are trademarks of Cloanto Italia srl. All other trademarks, property of their respective owners, are acknowledged. Cloanto has made every attempt to supply trademark information about manufacturers and their products. The following is a list of designations and their owners.

Amiga and the Commodore-Amiga logo are registered trademarks of ESCOM AG. ColorFont and ColorText are trademarks of Inter/Active Softworks. EGS 28/24 Spectrum is a trademark of Great Valley Products Inc. Fargo is a registered trademark of Fargo Electronics Inc. GIF is a trademark of CompuServe Inc., an H&R Block Company. PCX is a trademark of ZSoft Corp. Picasso is a trademark of Village Tronic Marketing GmbH. Piccolo and Rainbow are trademarks of Ingenieurbüro Helfrich. PostScript is a registered trademark of Adobe Systems Inc. Primera is a trademark of Fargo Electronics Inc. Retina is a trademark of MS MacroSystem Computer GmbH. Talon is a trademark of DKB. TIFF is a trademark of Aldus Corp.

For more information:

Cloanto

## 1.15 Cloanto

Cloanto - Company Information

Cloanto Italia srl	Tel +39 432 545902
PO Box 118	Fax +39 432 609051
33100 Udine	Bbs +39 432 545905
Italy	E-mail <info@cloanto.it>