Copyright © CopyrightÂ©1995-1997 Cloanto Italia srl

COLLABORATORS			

	TITLE : PNG		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		August 8, 2022	

REVISION HISTORY			

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

PNG

1.1   PNG DataType

PNG Toolkit and the PNG DataType v. 43.3, by Cloanto ^{1}

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 DataType for the Amiga, some background
information on PNG, sample source code to use DataTypes and an ARexx
script to automatically find GIFs and convert them to PNG.

The DataType supports all types of PNG-encoded images. Grayscale images
are converted to palette-based images. This version of the DataType
supports 24-bit DataType tag extensions, and can pass true color data to
supporting applications. In a palette-based environment, 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 DataType also reads the Author, Copyright and Comment fields
(DTA_ObjAuthor, DTA_ObjCopyright and DTA_ObjAnnotation DataType
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

DataTypes

Personal Paint

Cloanto

Legal Notes
^{1} The material presented here and the name Cloanto are ←
respectively
Copyright © 1995-1997 and a registered trademark of Cloanto Italia srl.

1.2 Installation

Installation

The PNG DataType 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 DataType 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 DataType, 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

                    CompuServe Announcement

                           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 Schalnat, 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 CompuServe Announcement

CompuServe Announces PNG-Based Graphics Specification
Fully Open 24 Bit Graphics Capability for Electronic Graphics Exchange

COLUMBUS, Ohio, June 15, 1995 -- CompuServe Incorporated today announced
the completion of a new 24-bit graphics specification that was announced
earlier this year. This new, enhanced 24-bit lossless specification will
offer the professional graphics community a significant enhancement to the
earlier GIF 89a specification while also eliminating the proprietary LZW
software, replacing it with compression technology compliant with the PNG
(pronounced 'ping') specification.

"The new specification is a true 24-bit lossless format that will give
users a 16 million color palette and represents a significant enhancement
over the previous GIF technology," said Tim Oren, CompuServe vice
president of future technology. "More importantly, this new specification
has been created with tremendous attention to making it free, open and
rights clear so that anyone can incorporate it into their products without

fear of patent infringement."

The new specification was developed as a collaboration between CompuServe and several key communities: The Internet PNG group led by Thomas Boutell, and including Jean-loup Gailly and Mark Adler, the developers of Deflate and Inflate; and the CompuServe online graphics forums (GO GRAPHICS). Ultimately, CompuServe's new graphics specification adopted compression technology that was based on the PNG specification. As a result of those efforts, CompuServe has determined that the PNG format closely meets the future requirements for graphics interchange on the Internet, on CompuServe and on other services. Based on current evaluation results, PNG will also be useful for exchange of information between graphics software products

"Earlier this year, there was a great deal of attention paid to GIF on the Internet," continued Oren. "Much of it was constructive and served not only to move the 24-bit graphics project off the back burner, but also gave us connections to the Internet team which helped us create the new PNG-compatible graphics specification in only five months. This cooperative effort has benefited the whole online community and should serve as a model for how the Internet's positive and creative forces can be focused."

PNG makes use of a data compression technology called 'deflation' used in the freeware Info-Zip programs. CompuServe has adopted the PNG format and is creating a free toolkit that will create graphics meeting the PNG specification while avoiding patent concerns. The toolkit will be available within the next few weeks.

Though CompuServe will hold a copyright on the toolkit, it is understood that its free distribution and use is encouraged and expected. To maintain the free and clear patent status of the new specification, it will not be backward compatible with the current GIF89a specification. Adoption of the new PNG-based specification will take place over time, allowing a smooth transition to the new format. CompuServe will also provide a conversion utility from GIF89a to PNG for use in conjunction with the CompuServe Information Services. This utility will be available within the next few weeks.

The CompuServe Information Service continues to be the world's most successful and most popular online and Internet service with millions of members who go online from more than 3.1 million active, paying accounts in more than 150 countries. The undisputed industry leader in innovation, the service offers global email, the industry's first CD-ROM supplement, libraries of free software, selected 28.8 kbps access and worldwide direct Internet access services. For a free introductory CompuServe membership, call 800-524-3388 and ask for representative number 664, or access CompuServe's home page on the World Wide Web (http://www.compuserve.com).

In addition to the CompuServe Information Service, CompuServe offers networking, Internet services, electronic mail and business information services to major corporations worldwide.

CompuServe is an H&R Block (NYSE: HRB) company.

1.6   From GIF to PNG

The GIF Controversy: A Software Developer's Perspective

An up-to-date version of this article is available at:

http://www.cloanto.com/users/mcb/19950127giflzw.html

1.7   PNG Specification

PNG (Portable Network Graphics) Specification

Please refer to the links in the following article:

http://www.cloanto.com/users/mcb/19950127giflzw.html

1.8   GIF to PNG ARexx Script

```
                /* GifToPng.pprx - Personal Paint Amiga Rexx script - Copyright ©  ←
                    1995-1997 Cloanto Italia srl

  $VER: GifToPng.pprx 1.1


                 More Comments
                */
IF ARG(1, EXISTS) THEN
    PARSE ARG PPPORT
ELSE
    PPPORT = 'PPAINT'

IF ~SHOW('P', PPPORT) THEN DO
    IF EXISTS('PPaint:PPaint') THEN DO
        ADDRESS COMMAND 'Run >NIL: PPaint:PPaint'
        DO 30 WHILE ~SHOW('P',PPPORT)
            ADDRESS COMMAND 'Wait >NIL: 1 SEC'
        END
    END
    ELSE DO
        SAY "Personal Paint could not be loaded."
        EXIT 10
    END
END

IF ~SHOW('P', PPPORT) THEN DO
    SAY 'Personal Paint Rexx port could not be opened'
    EXIT 10
END

ADDRESS VALUE PPPORT
OPTIONS RESULTS
```

```
OPTIONS FAILAT 10000

Get 'LANG'
IF RESULT = 1 THEN DO          /* Deutsch */
    txt_req_sel       = 'GifToPng-Zielverzeichnis'
    txt_err_svabort   = 'Speichervorgang wurde abgebrochen'
    txt_err_ldabort   = 'Ladevorgang wurde abgebrochen'
    txt_err_save      = 'Fehler beim Speichern: '
    txt_err_load      = 'Fehler beim Laden: '
    txt_err_oldclient = 'Für dieses Skript_ist eine neuere Version_von Personal  ←
        Paint erforderlich'
END
ELSE IF RESULT = 2 THEN DO  /* Italiano */
    txt_req_sel       = 'Selezionare cassetto'
    txt_err_svabort   = 'Scrittura annullata'
    txt_err_ldabort   = 'Lettura annullata'
    txt_err_save      = 'Errore nella scrittura: '
    txt_err_load      = 'Errore nella lettura: '
    txt_err_oldclient = 'Questa procedura richiede_una versione più recente_di  ←
        Personal Paint'
END
ELSE DO                /* English */
    txt_req_sel       = 'GifToPng target directory'
    txt_err_svabort   = 'User abort during save'
    txt_err_ldabort   = 'User abort during load'
    txt_err_save      = 'Error during save: '
    txt_err_load      = 'Error during load: '
    txt_err_oldclient = 'This script requires a newer_version of Personal Paint'
END


Version 'REXX'
IF RESULT < 7 THEN DO
    RequestNotify 'PROMPT "'txt_err_oldclient'"'
    EXIT 10
END


LockGUI
FreeBrush
IF RC = 0 THEN
    RequestPath '"'txt_req_sel'"'
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 '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
                    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
                                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 = txt_err_svabort
                                    ELSE
                                        errmess = txt_err_save || RC
                                END
                                FreeBrush 'FORCE'
                            END
                            ELSE DO
                                errcode = RC
                                IF RC = 5 THEN
                                    errmess = txt_err_ldabort
                                ELSE
                                    errmess = txt_err_load || RC
                            END
                        END
                    END
                    IF errcode > 0 THEN
                        BREAK
                END
                IF errcode > 0 THEN
                    RequestNotify 'PROMPT "'errmess'"'

                Set '"ICONS='iconmode'"'
                CALL CLOSE('listfile')
            END
            ADDRESS COMMAND 'Delete >NIL: 'tmpfname
END
UnlockGUI
```

1.9   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, etc.

   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 7.0 (PPaint Rexx version 7)
or higher, personal_png_io.library (enclosed with PPaint), and
personal_gif_io.library (or another GIF-compatible library).


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 (on the Personal Suite CD-ROM). A PNG
                DataType
                is available at no
cost for free electronic distribution.
```

1.10   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 dataype (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.

1.11   C Source Code

```
/*
   ViewDT

   Simple DataType-based Picture Viewer

   Syntax:
      ViewDT FILES/A

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

   Source Code Version:
      $VER: ViewDT.c 43.1
```

```
    Status:
        Public Domain

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

#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 (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 */
           {
              if (bmap != NULL && creg != NULL && ncol != 0)
              {
                 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, bmap))
                 {
                    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 = ERROR_REQUIRED_ARG_MISSING;
           }
           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);
```

```
        }
    }
}
```

1.12   Cloanto Amiga Software

                    Cloanto Amiga Software

For information on Cloanto Amiga products such as Personal Paint,
Amiga Forever, The Kara Collection and the Personal Suite, please refer
to the Amiga section on the Cloanto web site:

    http://www.cloanto.com/amiga/

For more information:

                    Distributors

                       Orders

                      Cloanto

1.13   Orders

                    Orders

For availability and pricing information regarding all Cloanto packages
and upgrades, please refer to the Amiga section on the Cloanto web site:

    http://www.cloanto.com/amiga/

For more information:

                    Distributors

                       Cloanto

1.14   Distributors

                    Cloanto Distributors

For distribution information please refer to the Amiga section on the
Cloanto web site:

http://www.cloanto.com/amiga/

For more information:

                    Orders

                    Cloanto

1.15   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 © 1995-1997 Cloanto Italia srl (Cloanto), unless otherwise
indicated in the files or in the following notice.

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 online distribution of the unmodified archive on the
Aminet, 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 personal use or online distribution by
Aminet and Cloanto require written permission or license from Cloanto.
The following section ("Specific Publications") contains a permission
valid for certain publications.

Specific Publications

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.com>) 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 Amiga Forever,
Personal Fonts Maker, PFM, Personal Paint, PPaint, Personal Write, PWrite,
Personal Suite, PSuite and The Kara collection 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 Amiga
International, Inc. ColorFont and ColorText are trademarks of
Inter/Active Softworks. CompuServe is a registered trademark, and GIF is
a trademark of CompuServe Inc., an H&R Block Company. EGS 28/24 Spectrum
is a trademark of Great Valley Products Inc. Fargo is a registered
trademark of Fargo Electronics Inc. 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. SBase Personal is a trademark of Oxxi Inc. Talon is a trademark of
DKB. TIFF is a trademark of Aldus Corp.

For more information:

             Cloanto

1.16   Cloanto

Cloanto - Contact Information

Cloanto   http://www.cloanto.com   E-mail: info@cloanto.com