Imagery v1.9

A utility to convert Macintosh, Apple IIgs, Atari ST, Amiga, IBM PC and Unix graphics, sound and movie files into Macintosh compatible files.

Sunday 19 June 1994

What's New In Imagery 1.9

Greetings. I'm sure some of you have wondered where Imagery had gone to - it has been almost two years since the last release. Alas, major work problems lead to almost no free time for me to work on Imagery. (This wouldn't be that bad a thing if I could say I made a lot of money doing all that work - but I didn't. Ah well.)

Anyway, Imagery should be back on track again and so without any further delay, here is Imagery 1.9.

New Features

Imagery now allows you to save a preferred conversion setup and allows you to convert batches automatically by drag-

dropping files over the Imagery icon while holding down the Command key.

Preview icons can now be added automatically by selecting an option in the output format selection dialog (see documentation).

EPSF files can include a Macintosh compatible preview as well.

IMPORTANT NOTE!
Imagery 1.9 requires System 7.0 or later and Quicktime 1.6.1 or later.

This version of Imagery requires Quicktime 1.6.1 or later for some of its features, and relies heavily on the offscreen bitmap features found in System 7.0 or later. Originally, my goal was to make sure that everyone, including users of System 6 and owners of 68000 based Mac, could convert graphics files on their choice of systems and machines. Alas, the sheer overpowering weight of new features and programmer support in System 7.0 and even more so in 7.1, and in Quicktime, which only works on 68020 and higher Macs compels me to move on.

The fact is that a new colour capable Macintosh is cheaper than your original MacPlus or SE and will give you many great new features. Similarly, System 7 is just too good an improvement over System 6 to put off using any longer. I know that System 7 eats up a lot more memory - try 7.5 with Quckdraw GX if you really want to see your memory vanish - but the truth is that you get a lot **for** that memory.

For owners of older Macs who cannot afford to upgrade - hey, I hear you and I know what you're going through. For what it's worth, this was not an easy decision for me to make. However, I want to get on with using the new features of System 7 and I don't want to have to write a whole version of MacOS to simulate the features I want to use in System 6.

The next version of Imagery will be even more tightly coupled to System 7, probably requiring features only found in System 7.1 or later, so be warned.

New Formats

Sounds and Movies. Imagery now supports converting several sound formats from various machines.Technically, this *is* outside the scope of Imagery which is supposed to be a graphics converter, but I got tired of all the strange sound converters which wouldn't just simply **convert** my file, but insisted on bringing up all sorts of issues. So, Imagery now simply converts sound files.

The sound file formats currently supported on input are:

Extension†	System	Format	Status	Description
.SND	IBM PC	Sound	Tested	SND Sound File.
.SOU	IBM PC	Sound	Tested	SOU Sound File.
.VOC	IBM PC	Sound	Tested	Creative Systems VOC Sound File.
'FSSD'	Macintosh	Sound	Tested	SoundWave Sound File
'jB1 '	Macintosh	Sound	Tested	SoundEdit Sound File
'Sd2f'	Macintosh	Sound	Tested	Sound Designer II Sound File
.AU	Unix	Sound	Tested	Sun/NeXT Sound File
.WAV	Windows	Sound	Tested	Windows WAV Sound File

.
The sound file formats currently supported on output are:

Extension†	System	Format	Status	Description
'snd '	Macintosh	Sound	Tested	System 7 Sound File
'AIFF'	Multisys	Sound	Tested	Apple AIFF File
'jB1 '	Macintosh	Sound	Tested	SoundEdit Sound File
.WAV	Windows	Sound	Tested	Windows WAV Sound File

Note that Extension refers to the file name extension or to the Macintosh file signature (type) code. Since Imagery automatically detects the format of Macintosh files, you shouldn't have to worry about this for Mac files, but for PC files, the filename is usually one to eight characters followed by a period and upto three more characters. The "period and upto three characters" are the extension for the file. On PCs, this is used to identify what kind of file the file is and Imagery also uses this as in many cases, it is the **only** way to tell what kind of file a file is.

More Bitmap Output Formats

Imagery now generates more bitmap output formats:

Extension†	System	Format	Status	Description
'EPSF'	Macintosh	Bitmap	Tested	Photoshop 2.5 compliant EPSF file with preview
'PICT'	Macintosh	Bitmap	Tested	Standard Macintosh PICT2 file
.PCX	IBM-PC	Bitmap	Tested	ZSoft PCX file
.TGA	IBM-PC	Bitmap	Tested	Targa file
.JPG	Multisys	Bitmap	Tested	JFIF standard JPEG compressed file
.GIF	Multisys	Bitmap	Tested	Compuserve GIF file

JPEG Support Integrated

Imagery now supports JPEG/JFIF file input and output directly within Imagery. No new versions of Imagery-JPEG will be released. **Note, this requires Quicktime and a Color Quickdraw capable Macintosh.**
Quicktime 1.6.1 requires a mere 20K of system heap when not in use and I argue that this is such a small overhead for so many features that **everyone** who can run it should have Quicktime 1.6.1 or later on their system.

Applications Files

If an application is dropped onto Imgery, it will find all 'ICON' and 'PICT' resources and convert them into the selected output format. This was added as a quick way to generate PC compatible files for porting Mac applications to Windows or to aid programmers in creating documentation for Mac applications.

Animation Files

These formats are supported differently. The Atari formats are treated as a sequence of bitmap images while the FLI/FLC files are actually treated as animation files and are converted to Quicktime. The Atari formats as well as Amiga ANIM and ANM files will be added to full Quicktime support in the next release.

Extension†	System	Format	Description
.ANI	Atari ST	Animation	Neochrome Animation File
.FLM	Atari ST	Animation	Animatic Film
.IC1	Atari ST	Animation	Imagic Film/Picture File (16 clr)
.IC2	Atari ST	Animation	Imagic Film/Picture File (4 clr)
.IC3	Atari ST	Animation	Imagic Film/Picture File (mono)
.FLC	IBM PC	Animation	FLI Autodesk Animation File.
.FLI	IBM PC	Animation	FLI Autodesk Animation File.

Bug Fixes

A bug which caused some GIF files to crash has been fixed.
A bug which caused the TIFF writer to create files which were incompatible with many TIFF readers seems to have been fixed.
More protection for memory overwrites and running out of memory has been added. A simple disk caching system has been added (this will be improved in the next release).
Problems with palettes in various modes have been fixed.

What's New in Version 1.8.5

New formats
Macintosh(tested)
Generic PICT/PICT2
All formats, restricted to current screen resolution.

Generic(tested)
TIFF 4/5 - all formats PC or Mac
All compression modes defined in the 8/8/88 version of the TIFF 5 manual are supported.

Fixes and improvements
Colour map errors for several modes should be fixed now.

Apologies
First to Phillip Wing for listing him as Paul Wing, and Jay van Vark for listing him as Jay van Ark in previous versions o
the manual.

What's New in Version 1.8

New formats
Macintosh(tested)
Photoshop (8BIM)
Grey, RGB and CMY/CMYK only. HSL, HSV and multichannel not supported.
Thunderscan (SCAN)
1, 4 and 5 bit files.
Imagestudio RIFF (Raster Image File Format)
Grey, RGB and CMY/CMYK only. HSL, HSV not supported.
"TPIC" Targa Files
Now map to ".TGA" files (see below).

IBM-PC(tested)
Targa (TGA)
Uncompressed modes have been tested. RLE not tested, Complex not supported.
Lotus PIC files
(Note, the handling of text in this mode is not the best... Thanks to Lotus for docs)
The type of ".PIC" is automatically detected.

Apple IIgs(tested)
Super-HiRes (.SHR) files.
Apple IIgs(partially tested)
Apple Preferred (.PNT) files. (Thanks to C. K. Haun)

Generic (tested)
"RAW" file import. See section on this mode as it is fairly complex.

Fixes and improvements
New file icons for generated files which more clearly identify the type of image saved.

Imagery now has a rather simple image viewing feature. If you select "Display" as an output format, the file will be converted then displayed. Please note, this feature works best under System 7 and may not work for 24bit files under System 6. Please note, this was thrown in more as a thought exercise and it's pretty lame... next version should be better offer more features.

IBM-PC PCX mode has been completely rewritten. Should work with just about everything.

IBM-PC BMP now handles more variants (Thanks to Jay Van Vark for these)
XWindows XWD pixel/byte order bug fixed. Should handle more images.
(Thanks to Jim McGowan for his samples)

Future enhancements
The next version will include Targa "TGA" and ZSoft "PCX" output. I have been told that a lot of "high-end" drawing programs for the PC can only support TGA files and that many other lower-end drawing programs can only support PCX For example, Windows' Paint program only supports BMP and PCX (which is all the stranger when you consider that Microsoft was one of the two original designers of TIFF...). I may also add an "IFF" output mode for Amiga users.

What's New in Version 1.7

Fixes and improvements
IBM PIC, GL and PCX format support improved. Fewer crashes.
(Thanks to Phillip Wing and Paul Jacoby for pointing out the problem)
Complete overhaul of memory checking done. Fewer crashes on unusual files. FIxes peculiar hehaviour on IIsi and IIcis.
(Thanks to Candi on AOL and Wally for their help.)
Amiga IFF-HAM colour translation now correct and ".HAM" or ".LBM" extensions are recognised.
(Thanks to Mike Monaco for his help)

New Features
PICT2 24bit files are now possible. Those formats which required that format, such as Amiga HAM files, now work. Also, all PICT2 files now use the "dither" copy mode which will let them translate better on monochrome and non colour Quickdraw Macs. Note, you require System 7 to see 24-bit direct colour images.

Known Problems
GIFLite
It seems that there is some sort of GIF "supercompressor" called GIFLite which is only available for the PC. It seems to compress GIF files upto an additional 25% and claims to keep the files compatible with standard GIF. Accordi to Paul Wing, some will work with Imagery but others crash seriously. I'm afraid that at this time, I have no intention to support this variant of GIF. If you do have problems with GIFLite files, try GIFConverter.

Lotus PIC Files
As I mention later in this document, MS-DOS tends to rely on a simple three letter extension to identify what the contents of a file are. Not surprisingly, some three combinations, such as "DOC", "TXT" and "PIC" get used a lot. Lotus and Pictor both use the "PIC" extension, but the files are very different inside. Imagery cannot read Lotus PIC files at th time, although I am attempting to locate documentation on this file format.

General Thanks
Thanks to Stu Gove for giving the Sun Raster section a good testing, to Gary Kessler for his excellent suggestions and o course, Paul Jacoby for poking where necessary when necessary.

What's New in Version 1.6

New formats
Unix (tested)
Sun Raster Files (.RAS)

Fixes and improvements
Critical failure in Atari ST PC1, PC2 and PC3 format conversion fixed.
(Thanks to Ewen Wannop for pointing out the problem)
Some GRASP GL files would cause lockup and bizarre offsets - fixed.
(Thanks to Paul Jacoby and Harry@UMSLVMA.UMSL.EDU for their help)

What's New in Version 1.5

New formats

IBM-PC (untested)

- QRT Raytrace (.QRT) files
- MTV Raytrace (.MTV) files

IBM-PC (tested)

- .IMG DRI GEM Image files (mono & colour)

Unix (tested)

- XWindows Window Dump Files
  - (.XWD - Thanks to Michael Garnett for the sample file)
- XWindows XBitmap Files (.XBM - Thanks to Mike at NCSU for the generator program)
- Portable Bitmap (.PBM .PGM .PPM) files

Unix (untested)

- Sun Raster (.RAS) files
- CMU Window Manager Dump Files (.CMU)
- MGR Window Manager Dump Files (.MGR)
- FITS Files (.FIT)
- HIPS Files (.HIP)
- Usenet FaceSaver Files (.FSV)
- Xerox Doodle Brush File (.XDO)

Additions to existing formats

Amiga IFF Extra-halfbright mode is now supported. (Untested)

Fixes and improvements

PIC decoder could hang on improperly created files and colours were sometimes incorrectly translated. These problems have been fixed.

Previous versions of Imagery would allow you to select an output format which has fewer bits per pixel then the current image. In fact, Imagery doesn't support this yet with the result that the output file contains gibberish. For example: if the original file was a four bit palette image and the output was TIFF B/G, the output file would be a four bit grey image, bu no mapping for the colours would take place - the image would look strange. Similarly, a 24 bit RGB image would gene total garbage if TIFF P or TIFF B/G were selected. However, most of the 24 bit RGB modes override the user selection, this has not been seen too often.

This version of Imagery is smarter about such things and will not let you select an inappropriate mode, forcing the selec to the lowest valid one. There is no way to generate a TIFF-B monochrome bitmap from a grey or colour image in this version. That feature will be available in v2.0 which is slated for release at the end of March or April 1992.

New features

In answer the the many requests for PICT2 and GIF output, Imagery now offers these output formats. PICT1 (old pict) i not supported, but since System 6.0.5, all Macs including those without colour quickdraw have been able to decode colo PICT2 files. GIF files are always interlaced - since the primary use for GIF is to upload images, they should be interlace this lets a downloader preview the image in less than 1/8th the file's full download time. Imagery does not support GIF f of more than eight bit (256 colour) depth nor PICT2 files for 24 or 32 bit colour (yet).

What's New in Version 1.2

New formats

IBM-PC EGAPaint/ColoRIX RIX files (untested)

Additions to existing formats
Atari ST DR Doodle (DOO) and STad PAC format's now tested
(Thanks to Wolfgang Lang for the sample files)

Fixes and improvements
Better on-line file support documentation
Monochome TIFF-P palette now automatically generated

What's New in Version 1.1

New formats
Compuserve's GIF
Compuserve RLE and C-RLE
IBM Targa files (modes 0, 1, 2, 3, 9, 10 and 11) (untested)
Atari FCP (Flash Colour Picture) version of RLE
Atari IMG files (IBM IMG files not supported yet)
- old format, STTT and XIMG formats supported

Additions to existing formats
Window's BMP compressed files (BMP4 and BMP8)
Amiga IFF RGBN and RGB8 files

Fixes and improvements
IBM GRASP control file now written as TEXT file
IBM GRASP image files are now automatically converted to TIFF
User can now select destination directory

Abstract

Imagery is a simple utility which converts a range of graphics files from the IBM PC, Macintosh, Apple IIgs, Atari ST, Unix and Amiga computers. The program generates standard TIFF files in TIFF 4.0/5.0 format compatible with Freehan Digital Darkroom, Superpaint and many other drawing and editing programs.

Introduction

Like many people who wanted a Macintosh but couldn't afford one, in 1985 I bought an Atari ST - one of the first in Canada and the very first one in Alberta. Within a year, Dave Small created a gimmick called "MagicSac" which allowe an ST to work like an old 64K ROM version Mac.

I managed to get a lot done on a single 400K drive system with about 400K free memory. Later I added a second 800K drive. However, when Small left his company to form a new company which created a 128K ROM version of MagicSac called Spectre 128 (later Spectre GCR), I found myself with an interesting problem.

To upgrade my ST to the point where it could be seriously useful with Spectre, not to mention buying the Spectre and obtaining a legal set of Mac 128K ROMs - which was becoming increasingly difficult, I would end up spending almost as much as buying a brand new Macintosh SE.

Also, Atari had a bad upgrade record with me. My original machine did not have ROMs, although the advertising guarenteed them. Same for an internal RF modulator. Atari never offered the RF modulator as an upgrade and only offer the ROMs as a $30 upgrade. Not much, I appreciate, but I had already more than paid for them. Well, when I asked Atari how I could upgrade my Atari ST to STm level (let alone STe), I was told "You'll have to buy a new machine" - so I did Macintosh.

However, I was stuck with a **lot** of Atari image files in five or six formats (Degas, Degas Compressed, Neochrome, NVi and a handful of random singles) none of which were compatible with each other, let alone with anything on the Mac.

So, I wrote a program then called “TIFF-ST” which converted all of these various formats into TIFF files. I chose TIFF a couple of reasons. First, it is a standard and an easily implimented one. It’s not proprietory. There is no special comple compression system unless you choose to use it and virtually all Mac graphics programs can use a version of TIFF.

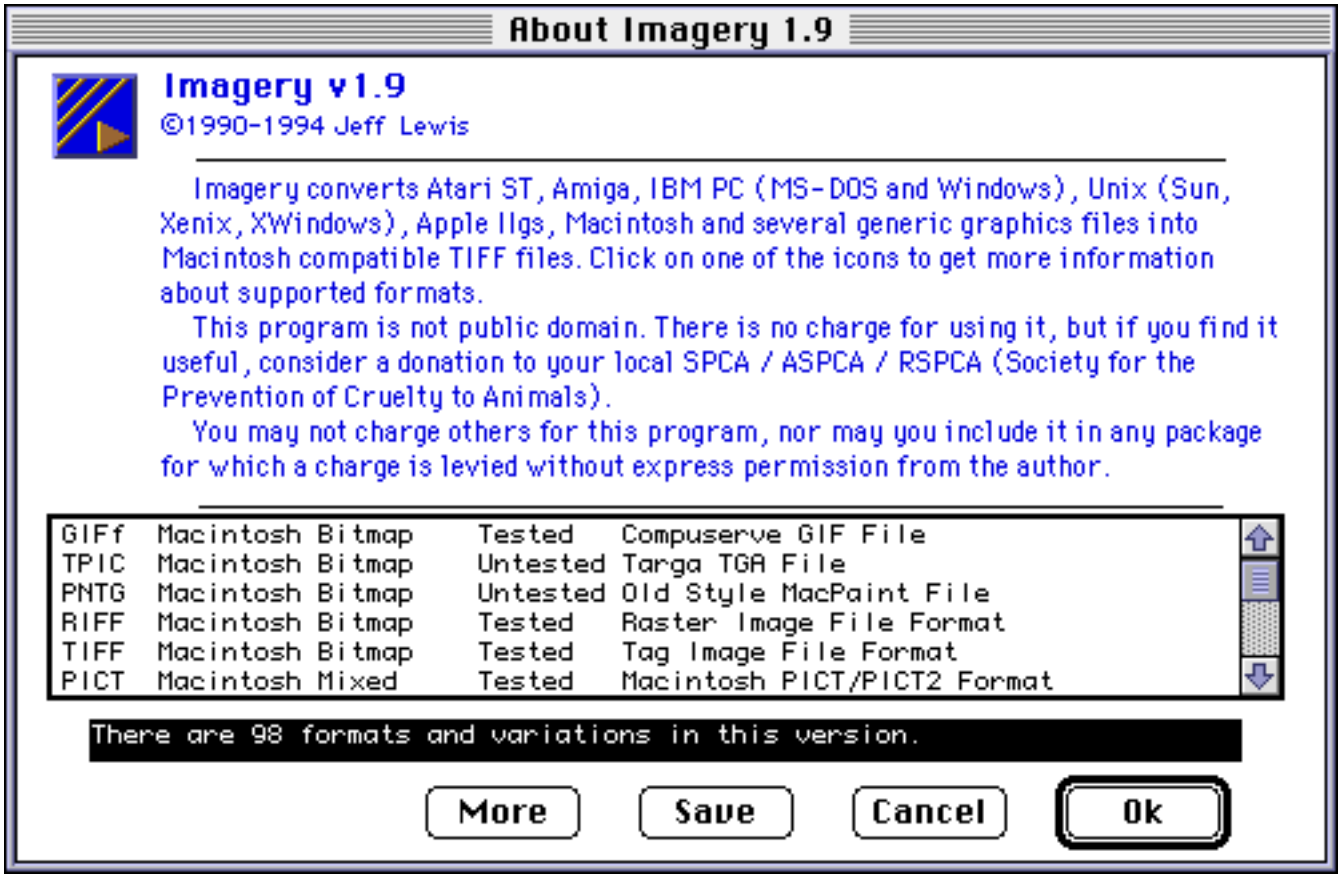
Later, I found I was getting Amiga IFF files as well as a variety of IBM PC graphics files such as GRASP and PCX. Wit Windows, BMP was added to the list. I added these to the original TIFF-ST program and changed its name to Imagery.

Since then, I have spent a quite a bit of time and effort gathering nearly 4.3MB of information on every file format I can find. I am compiling these files into a single document which I hope to post or publish sometime early next year. Many these graphics formats have been added to Imagery and I am still adding more with the addition of PICT based images t handle a
collection of vector based image formats such as DXF, IGES and CGM to occur in version 2.0.

Usage

Imagery is very simple to use and under System 7 can be used in two different ways.

The simplest way is to launch the program by double clicking the Imagery icon or by selecting the Imagery icon then selecting Open... from the File menu. The first dialog box is a copyright notice which can be dismissed by hitting Enter Return or by clicking on "Ok".



The biggest new feature here is the replacement of the multiple system specific panels in previous version which lis supported formats and their status, with a single scrolling list box and a notice indicating the number of formats and variants supported in this version of Imagery. A variation of a format typically is a format which has more than one extension - the worst offender being Sun .RAS files which are also known as .RF, .RS, .IM1, .IM8 and .IM24 files,

counting for six of the formats and variations.

The first column is the file extension or file type code. Extensions are used on PCs and other computers to identify t type of file and usually are several letters preceeded by a period placed at the end of a file name. For example, if the file's name is "bigpict.bmp", the ".bmp" part is the extension and indicates (if this is from an IBM PC or compatible that the files is (or rather is likely) a Windows BMP file. I say "is likely" because with a limit of eight characters, the period and three characters for an extension, some of the more obvious extensions tend to get used by several differe programs for completely different files. Worse, in many cases, there is **no** way to tell these variants apart.

On the Mac, each file is marked with a signature, a four letter code which identifies the program which created the f and a type, another four letter code which represents what's in the file. Since both upper and lower characters are distinct, there are 2,820,802,151 file types (not every character is allowed). As well, Apple asks each developer to register their types and signatures to try and prevent collisions. While some have happened, in over ten years, very f have.

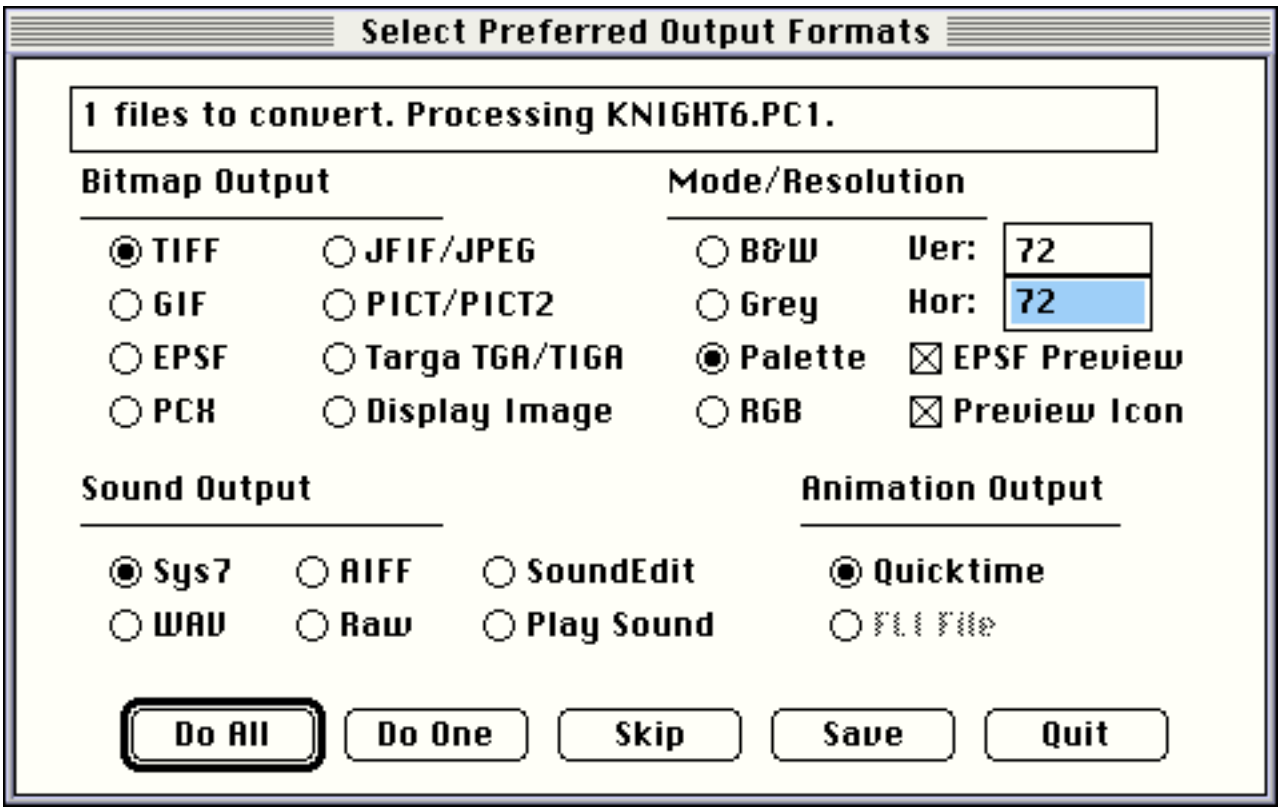
The second column identifies which system the file is most likely to be found on. The third identifies the type of file bitmap, vector, sound, animation or mixed format file. The next column gives a status on the format to indicate if it' been tested, is untested, buggy or is known to fail. "Untested" does not mean "Doesn't work". What has happened is that Imagery supports many formats which are so obscure that I could not find a sample file for that format. So, if m documentation is right (and I got the code right) they should work.

The final column simply gives a description of the file type.

Along the bottom you'll see several buttons. The **OK** and **Cancel** buttons are obvious. **OK** continues into the progra while **Cancel** quits the program immediately. **Save** will save a text file of the formats listed in the box above the buttons and **More** gives you more copyright information of interest mostly to publishers and people who want to include Imagery with their collections.

Selecting Output Format

Selecting "Ok" for either of the previous two dialogs will take to you the output file format selection dialog shown below.



If you're familiar with earlier versions of Imagery, you'll notice a lot of new options. First, since Imagery can handl sound and animation files, some way to handle selections of output formats appropriate for those kinds of files was necessary. Worse, since you can drag drop a collection of files with sounds, bitmaps and movies all mixed together, simultaneous selections were necessary.

Fortunately, it's not that difficult. The panel at the top of the dialog will tell you how many files are left to process a the name of the next file to process. **Bitmap Output** allows the selection of output format for bitmap files.
Mode/Resolution options modify the output format for bitmaps where appropriate by allowing you to select preferr file type (monochrome, greyscale, palette or full 32 bit RGB - Imagery does not support 16 bit output at this time), preferred resolution, whether or not to add a Macintosh preview to EPSF files and whether or not to add a preview i to the file (System 7 only).

JPEG output requires Quicktime. If Quicktime is not detected, this option will be dimmed.

Display output brings up a window with the image displayed in it. You can continue from this window by clicking the close box, or by hitting the space bar.

EPSF files are Adobe Photoshop 2.5.1 compliant and seems to work correctly with Freehand 4.0 and other programs which can handle an EPSF file. They are also Adobe EPSF 3.0 compliant and are generated in ASCII format as recommended by the Adobe Redbook on Postscript, version 2.

You'll note I stated that these are 'preferred' formats and option because in some cases, Imagery will override your choice and select a more appropriate format. For example, Lotus PIC files are vector files and selecting a bitmap for will be overridden in favour of PICT files. A future version will simply convert the PIC to the selected bitmap forma probably v1.9.1.

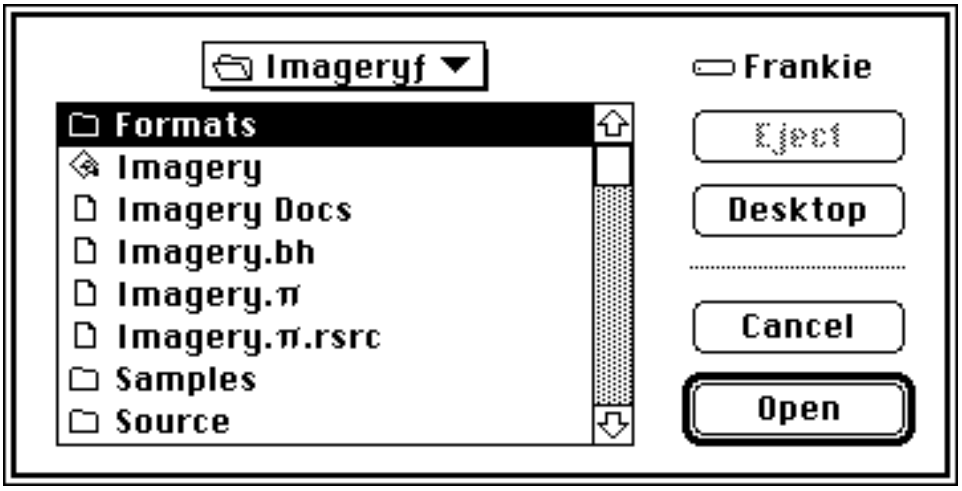
Sound Output allows you to select the preferred output format for sound files that are detected.
Movie Output allows you to select the preferred output format for animation and movie files which are detected. A the moment, only Quicktime MOOV files can be generated and require Quicktime. If Quicktime is not detected, this option will be dimmed.

Unlike previous versions of Imagery, this version will allow you to convert files one at a time, changing the output options for each file, or all at once using the same set of options for all files.

The **Do All** button will use the current settings on all files either drag-dropped or selected one at a time. If this butto pressed, you will not see the output format selection dialog again during this session. The **Do One** button will apply current settings to the next file and then ask for new settings. **Skip** simply skips the next file. **Save** saves the curren settings into the Imagery program. These settings will be used as defaults for output format from the time you save them onwards. As well, if you drag and drop a set of files onto Imagery while holding the Command key, the curren saved settings will be applied to all files automatically. This prevents any dialogs from being shown. **Quit** exits Imagery immediately.

Selecting Input

The next dialog will be the Standard File Selector and you may repeated select image files until all conversions are complete. Selecting "Cancel" will end the program. As each file is processed a status window will inform you about file being processed and of the current status.

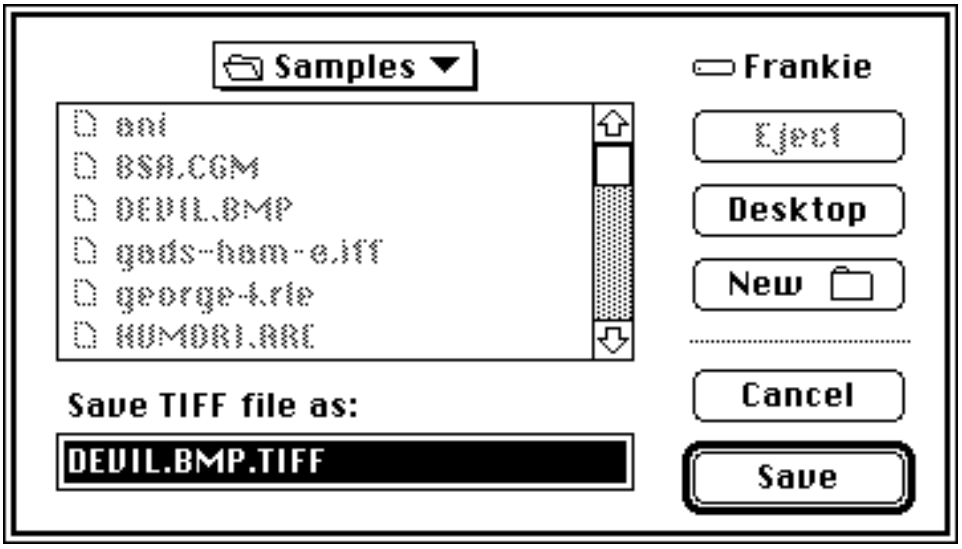


Since the image files will all be coming from non-Macintosh systems which do not use the Mac's file signature syste there is a problem identifying a graphic file and its internal format. On other systems, a unique - well, mostly unique extension, typically of three letters, is used to identify the file type. I say "mostly unique" because in fact, there are several major cases where wildly differing file formats have the same extension.

To make things simpler, Imagery looks at the file's extension just as a program on one of the other computer systems would and then presumes the image format. Many of the translators also do several checks on the file to confirm tha the format matches the extension, but many either do not or cannot (Atari files are the worst for this). The previous list will give you the required extensions for all formats currently supported.

Saving Output

The program will analyse each file in turn and then generate a new file by appending ".TIFF" to the end of the file a assigning it a "TIFF" signature. This will allow most program which can recognise the TIFF type to see these genera files. A standard output file selector will be displayed:

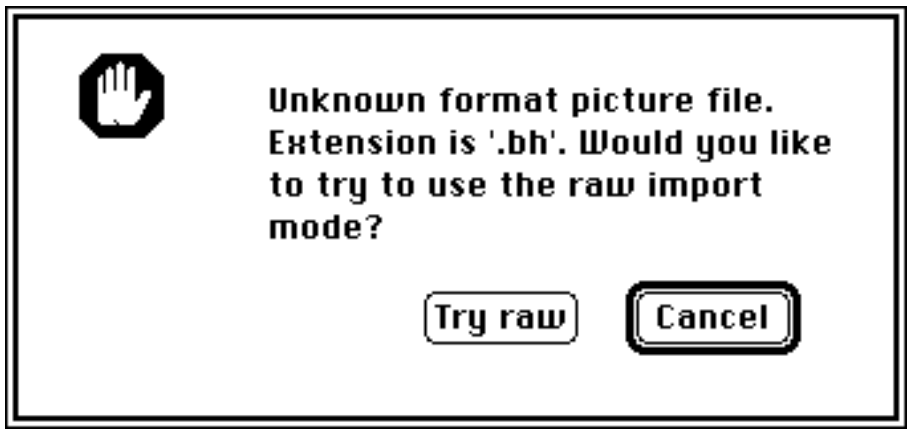


Animation files, or composite format files such as GRASP will generate multiple files. If there is a mechanism to identify the files within the source file, then the files will be given those names. Otherwise, they will take on the sou file's name with a number appended to it.

With these files, you specify the destination for the first file and then that file and all other components will be saved into the same place.

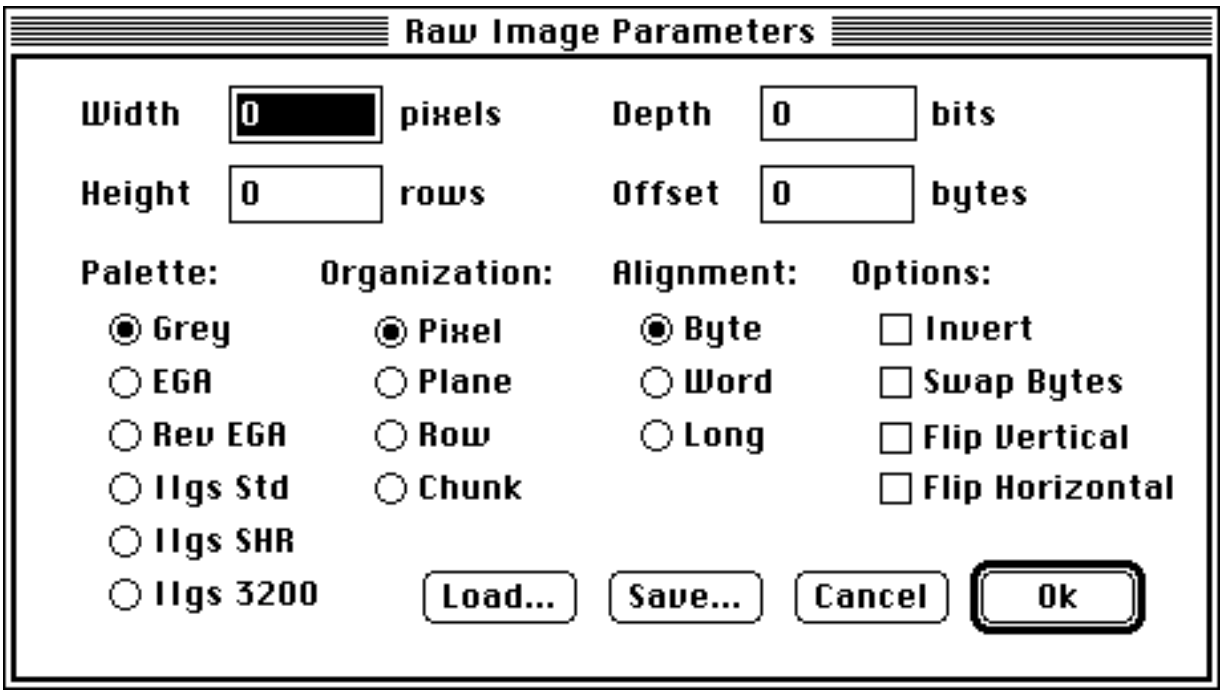
Raw File Import

Imagery now includes a "raw" file import. This allows a knowledgable user to try and decypher and extract a usable image from an unknown or damaged file. Unfortunately, there are so many possible file formats and file compressio methods that not all possible files can be imported using this module. Also, the number of parameters and variations makes using this more complex than would be desired.



When a file cannot be recognised, the user will be given a chance to try raw import. A file extension of ".RAW" will automatically select this mode. You should not select this unless you have some idea as to the internal format of the No damage to the original file can happen by selecting this mode.

The raw image control dialogue looks like this:



Told you it was complex, but don't despair, it's not as bad as it looks. Let's examine each of the fields and button groups.

The first four fields are pretty straight forward. Width and height define the picture size in pixels and rows. The dept field defines how many bits are used to define each pixel. Offset defines how many bytes to skip from the start of th file to the start of the image data.

Beneath these are three groups of radio buttons and one group of checkboxes.

Palette: This attempts to define a palette to translate the pixel data into colours. However, the decyphering of colou palette information (when available) is extremely complex and so, for this version, it is limited to:

Grey
: This is the same as having no colour translation.

IBM-PC EGA mode (Atari ST default 16 colour mode)
: Normal EGA for bitmaps stored as IRGB.

Reverse EGA mode
: Used when the EGA bitmaps are stored in reverse order (BGRI)

IIgs Standard, IIgs SHR, IIgs 3200
: Apple IIgs standard palettes.

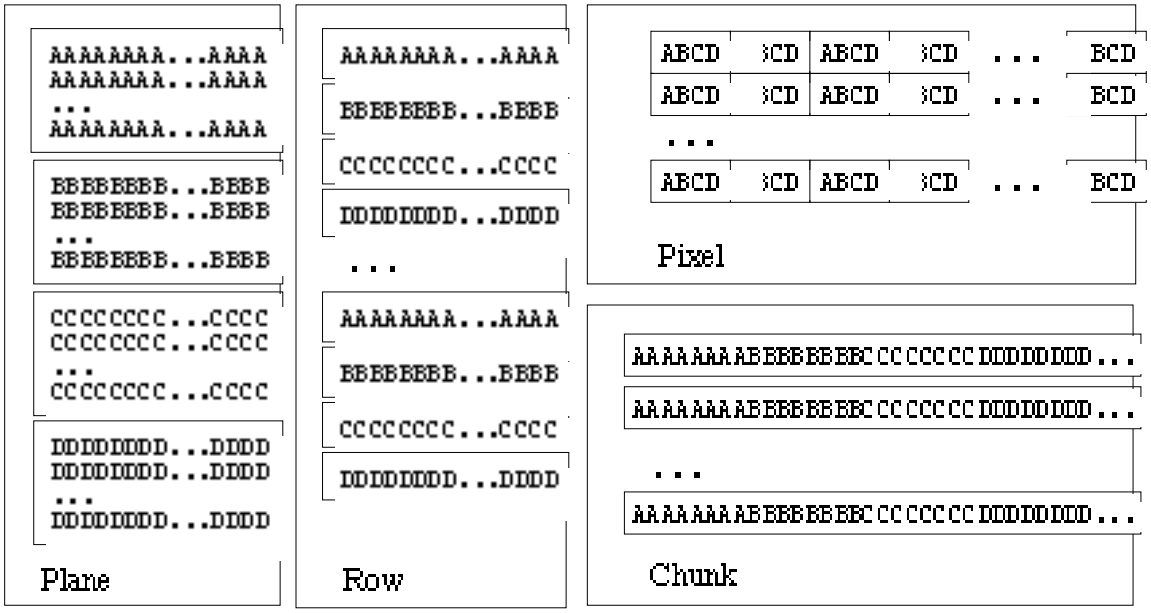
Organisation: This is the most complicated part of the raw import. Files can be arranged in many different ways. So keep each pixel's data in one chunk, others gather all of the most significant bits in one plane, then the next and so o

Let's look at one pixel of four bits

A B C D

where "A" is the most significant bit (causes the biggest change in colour or grey leve), and "D" is the least signfica bit (causes the smallest change in colour or grey level). Our hypothetical image can be thought of being made up of *width* pixels each of four bits as above and *height* rows of pixels.

Four specific organisations are allowed in this import:



Pixel

Each pixel of the image is complete with all bits following from most to least significant. If the number of bits (dept is not an exact multiple of 2 (1, 2, 4, 8) then the bits will be packed across bytes.

Plane

The bits which make up the pixels are gathered into planes of similarly significant bits. For example: all the most significant bits are gathered together as if they were a one bit depth image followed by the next to most significant so on until the least significant bit.

Row

Similar to the *plane* format, but the bits which make up the first full row of the image are placed one following the other with the next following and so on.

Chunk

This is the most complex format. Based on the *row* format, it further breaks up the mage into chunks which are interleaved. For eaxample, on the Atari ST, you will see the most significant bits for the first sixteen pixels in the fir word, the next most significant bits for the first sixteen bits in the next word and so on down to the least significant for the first sixteen pixels in the last word of this group. This then is repeated for the next sixteen pixels and so on fo the rest of the image. Imagery takes the size of the chunk from the alignment setting (see below).

Alignment: All data in a file must be aligned such that it takes up an exact multiple of eight bits - one byte. For example: even if each row contains six one-bit pixels, you can't write **just** six bits, you have to write a byte. This is "byte aligned". However, some systems, such as Microsoft Window's BMP, require alignment to a larger chunk - lon (32bits) in the case of BMP. This options lets you select the correct alignment. **Byte** aligns to the next 8 bits, **word** t the next 16 and **long** to the next 32.

Also, if the image's organisation is "chunk", then this setting also determines the size of the chunk.

Options:

Invert

This turns the image into a "negative" of the data. Some computers consider the lowest number (usually zero) to be black and the largest number to be white while other are the other way around.

Swap Bytes

This option reverses the chunks of the image from Motorola to Intel byte order. The chunk size is determined by the alignment setting (see above).

Byte: Not affected.
Word: AB -> BA
Long: ABCD -> DCBA

Flip Horiztonal
This option flips the image from left to right.

Flip Vertical
This option flips the image from top to bottom.

System 7 Support

This version of Imagery has minimal support for System 7 other than to ensure it is compatible, but it does take into account one feature of the System 7 Finder to make using Imagery simpler.

Under System 7, you can drop-launch a program by dragging a file of a type known by that program over the icon f the program. A similar mechanism exists in System 6 and earlier by selecting multiple files including the Imagery program, releasing the shift key and then double-clicking the program icon.

Imagery allows any file type to be dragged over its icon and also allows as many files as you wish to be selected thi way. When you drop-launch Imagery, the first copyright screen is skipped, as is the Standard File Selector for select files and specifying where you want the file to be saved. Only the TIFF format selection dialog will come up to sele the format for all the files in the selection.

A future version will also add AppleEvents to allow other programs and scripting systems to convert a file on the fly with no user interface at all - the Mac equivalent of a Unix filter program.

Special Features

GRASP
In version 1.0 of Imagery, GRASP GL files were unpacked into a collection of PIC and CLP image files and a TEX control file. In version 0.5, the TEXT file was given a binary file type which made editing it difficult. As well, the te file was not converted from MS-DOS format to Mac format.

In this version, the PIC and CLP image files are now automatically converted to TIFF and the control file is now a TEXT/MACA file which can be opened with TeachText or any other editor which can read standard TEXT files. Th MS-DOS format for this text file is now converted to Macintosh format.

What are TIFF Files?

TIFF stands for "Tagged Image File Format" which was developed by Aldus (Pagemaker, Freehand, Superpaint and Digital Darkroom) and Microsoft (Word, Works, Project, File, etc.). It is a machine independent format designed exclusively for bitmap images.

What makes TIFF unique is its tag based system. Other file formats such as the Amiga IFF and Compuserve GIF formats use tags, but neither of these use a directory system to make finding the tags quick and simple, nor do they allow the highly flexible format and extension of the file format that TIFF provides.

TIFF files currently can handle simple monochrome bitmaps, palette based colour images and RGB full colour imag of unlimited colour depth (32bits of resolution specification per colour - 4Gig bits of colour info per colour!). As we there are four compressions schemes: no compression, RLE/Packbits (MacPaint) compression, CCITT-3 Huffman compression (FAX) and LZW (Lempel-Ziv Welch) compression similar to the one used in GIF.

Many PC graphics programs and DTP programs can now support TIFF and several use TIFF as a default format for

saving pictures. Similarly, most Mac drawing programs and DTP packages can import and export TIFF format files. you aren't sure, check under "Import" for your favourite drawing or DTP package to see if it is supported. Several programs on the PC now use TIFF exclusively for their file storage.

While TIFF files cannot support object based images without having them first translated into a bitmap image, they very effective for storing scanned images (which was the original intent of the file format) as well as complex and h resolution colour drawings. Further, since the format for this file is widely distributed and is open-license, it is easy anyone who wishes to support this format to add it to existing software. In fact, several companies including Hewlet Packard and DEST have been distributing low cost developer's kits for TIFF for several years.

TIFF Compliance

The TIFF files Imagery generates comply very strictly to the TIFF 5.0 specs as defined in the Aldus/Microsoft TIFF Standard Specification version 5.0 final dated 8 August 1988, with all defaults selected to allow maximum compatibility with TIFF 4.0. The only exception is that I use the *SubfileType* tag instead of the recommended *NewSubfileType* because either will work and the former is still more widely accepted. Similarly, wherever the TIFF document mentions that TIFF 4.0 only allowed one of several choices now offered in TIFF 5.0, I have selected that choice to ensure compatibility with older TIFF 4.0 readers.

The writer conforms to TIFF X, R, G, B and P standards as defined in Appendix B of the aforementioned document includes all required tags for those formats.

None of the generated TIFF files use any form of compression. This is because most Mac and PC programs are still based on TIFF 4.0. TIFF 5.0 adds LZW compression similar to the one used in GIF files, but many readers still can support that. When TIFF 5.0 becomes the dominant form, I'll add the compression.

The “TIFF Incompatibility Myth”

One of the most annoying statements to me it the one that goes “Yeah, but the real problem with TIFF is that there a so many incompatible kinds of files...” Well, I’d like to take a moment here to debunk that myth.

One aspect of TIFF files is that there are both standard and non-standard tags. The standard ones are clearly defined the TIFF specifications, but the specs also allow for developers to create and use their own tags. Tag numbers zero t 32,767 are reserved for standard tags while the “negative” tags (-1 to -32787) are reserved for developers. Further, Microsoft and Aldus are supposed to keep a complete list of assigned tags. What’s supposed to happen is that a developer writes in and requests a block of tags for their own use - they don’t even have to say what the tags are for.

Ok, sounds like a problem, right? How does a reader know what to do with these developer’s tags when there is no w to find out what a tag does?

Well, the answer is this - you don’t have to know. In theory, the image in the file should have all the standard tags ar the developer’s tags contain length information. If the tags are correctly formed and written, and if the reader is writ correctly, everything should work fine. The image you get may not be perfect, especially if the developer tags conta information about modifying the image, but at least you can get the image out and then reprocess it yourself.

So, why the incompatibility myth?

This comes from two things that stem from two all too common problem - programmer laziness and programmer arrogance. The first happened on the IBM PC. The TIFF specs clearly state that **all** TIFF readers should be able to re MM (Motorola byte order) and II (Intel byte order) files. Most Macintosh applications which could handle TIFF file fact did read both forms. However, few PC program would read MM format files. This is laziness **and** arrogance.

The second problem lay with programmers who simply could not stick to the standard. By this I refer to two things:

creating new tags without registering them and worse, not following the specs when writing the standard tags. An excellent example of this is Teletypesetting's T-Script PostScript rasterising program. Their 1.4 TIFF export module ignored the rule that states that tags **must** be written in tag number sorted order. Theirs doesn't. Smarter programs (li Aldus's Freehand program which seems to be able to read correctly just about anything as a TIFF file) will presort th tags anyway just to make sure - others don't and reject the file.

(The people at Teletypesetting claim that they were working with a newer version of the TIFF standard which allow non-sort order, but I contacted the TIFF support desk at Aldus and they assure me that the 8/8/88 document defining TIFF 5.0 is still the most recent version and that it requires tag sort order. Version 2.0 of TScript fixes this problem.)

Simply put, when the TIFF 5.0 spec is followed to the letter, by programmers writing both readers and writers of TI files, there is never a case where any TIFF file will be incompatible with any reader on any CPU, except in cases wh the program in question is not capable of working with the data contained within the TIFF file - a monochrome bitm drawing program would not be able to handle TIFF-G, TIFF-P or TIFF-R files all of which contain colour or grey images.

The only other area of contention is the recent addition of LZW compression. Most readers are still TIFF 4.0 readers which are mostly compatible with TIFF 5.0 files except in two ways - one critical tag has been changed (SubfileTyp NewSubfileType) and LZW compression was added. This is why Imagery generates TIFF 4/5 intermediate files. It c not use LZW and it uses SubfileType tags - which is allowed under TIFF 5.0. Imagery files should be readable by TI 4.0 and TIFF 5.0 readers.

I hope this puts an end to the incompatibility myth.

TIFF 6.0

Aldus, who currently seem to be the driving force behind TIFF, are about to release version six of the TIFF specs an there are many new and very interesting features including JPEG support. However, much as LZW compression is s widely unused, almost four years after its introduction to TIFF, I suspect that it may be many years before these feat become widely used. This is a great pity because they really have added some impressive new features.

However, until TIFF 6 becomes more widely supported by readers, Imagery will continue to generate TIFF 4/5 compatible files. The upcoming "Pro" version will offer selectable TIFF 4, 5 or 6 mode generation.

User Interface Issues

A couple of Imagery users have written to me about my choice of a user interface. Specifically, they argue that it's n Macintosh-like enough. Two points on that: first, check out Font/DA Mover. Written by Apple, part of the system software package... and it works very much like Imagery.

Second, and more importantly, my idea was to create something which is very commonplace in Unix - a filter. In Un it is common to take a file and pass it though a chain of small programs, each of which do one thing, modifying the and passing it on to the next program in the chain with the final file being written at the end.

This isn't possible on the Mac, but with the addtion of drop-launching, I tried to create a similar idea. You drop the f on the first icon and then pass it through each one until you've filtered the file into a desired form.

The fact is that currently, Imagery isn't graphically oriented. To add menus and windows would make the program le effective, not more so. However, Imagery Pro will include editing and modification windows and will gain a user interface when run directly. It will, however, still include the same drop-launch conversion mode.

I hope this clears up the questions.

Copyrights and Fees

The author may be contacted via Compuserve at 76217.2241 or via the Internet at werewolf@netcom.com.

Bugs and Warranties

Warranties: there are none. This is a free program - what do you want for nothing? :-) You use this at your own risk. tested it as well as I can and while it's not the best thing I've ever written (although it is getting better) - it is simple, clean and functional.

Bugs: there are probably plenty. I have tried to catch as many as I can, but this was a hack utility which has become of a project/hobby and will grow on forever simply because I enjoy playing with it.

Known bugs and weaknesses

In order to make the odd file which aligns a row to a long word rather than to a byte (Windows BMP files for examp fit in with TIFF which is row/byte aligned, I just redefine the image to be a little wider. This results in several pixels garbage along the right edge. It's unlikely this one will be fixed soon.

Some file types outright lie about the real size of an image. In these cases, the program may crash because it overrun the image buffer. Some formats are more susceptible to this than others and with these, more extensive checking is done.

Amiga HAM-E IFF files are not yet supported. Normal HAM and Extrahalfbright is supported.

The GRASP decoder simply breaks the GL file into its constituant files. These are PIC files which are then quietly rerun through Imagery to convert them to TIFFs. The control file has a TXT extension and can be edited with a text editor.

None of the animation decoders actually generates an animation file. Until a standard for animation is defined (ie: QuickTime), this is the best I can do. Palette rotation animation is not supported at all. As well, GRASP files contain control program which is interpreted and that is outside the scope of Imagery.

I do have an interpreter shell written, but when I asked the author of the PC GRASP program for permission to creat Mac interpreter, he didn't think much of the idea and he did not actually give permission to do this. Since GRASP is copyrighted program, I do not wish to get into legal hassles over this. So, I am in the process of writing an interpreter/translator which will "compile" the GRASP file into either a PICS file, a MooV (QuickTime movie file) into a composite file which can be interpreted by a to-be-written run-time system. Since this new form would not actually be interpreting GRASP commands, this should neatly avoid any legal issues.

Unfortunately, it is unlikely that such a feature will become available until after version 2.0 is realeased.

(Note: Since the writing of the above, several people have pointed me to existing GRASP viewers for Unix based or Windows. I have obtained the sources for these viewers and hope to be releasing a GRASP viewer called MacGrasp shortly.)

Comments and Reports

I can be reached via Compuserve at 76217,2241 or via the Internet at werewolf@netcom.com. I'd like to hear about any bugs you may find, and if you know of a graphics file format Imagery doesn't support let me know. If you have layout for the format, please include it and I'll write it in as quickly as possible. If you have any suggestions, I'd also like to hear them, as long as they're polite :-).

Please note that I am no longer reachable via America Online. To explain why, I live in Canada and while America Online is one of the least expensive data services in the States, it is the most expensive one in Canada because of a rather larger service charge they levy. For example, Compuserve (which is the cheapest service in Canada, BTW) ru around CDN$12/hr for 2400 baud connections, while AOL costs US$18.50/hr. I reget this, but until AOL brings thei costs into line with CIS in Canada, I cannot afford to use AOL. In fact, it would be cheaper for me to call AOL in th States after 6pm by long distance than to use it in Canada.

If you have a file which will not convert correctly, you can send it to me at either address, although I would prefer Internet if possible. I can also make an FTP site available if you need to send a large file. Please email me if you require this.