

**Name:** ImageViewer (Version 0.9k)

**Category:** Internet

**Description:** ImageViewer is a little utility that allows you to view images stored in a number of different formats. The images can be arbitrarily scaled and optionally saved in any of the write supported formats. The app also provides an image filtering service that will convert any of the listed image formats to TIFF. Just make sure that ImageViewer is installed somewhere where Workspace knows about it (eg. ~/Apps) and it will kick in automatically.

**Price:** Public Domain

**Availability:** m68k, intel, hppa, sparc

ImageViewer is a freeware application that gives you the ability to display and convert between a multitude of image formats. Among the supported formats are:

atkras	Andrew Rasterfiles (binary and ascii)	read		
gif	Graphics Interchange Format	read		
itex	ITEX Framegrabber Files		read	write
pbm	Jef Poskanzer's Portable Bitmaps (binary)	read	write	
pgm	Jef Poskanzer's Portable Graymaps (binary)	read	write	
ppm	Jef Poskanzer's Portable Pixmap (binary)	read	write	
jpg	JPEG File Interchange Format (JFIF)	read	write	
macp	MacPaint Documents		read	write
mtek	Mikrotek Scanner Files		read	
eps	PostScript Encapsulated Files	read	write	
ps	PostScript (plain)		write	
ras	Sun Rasterfiles		read	write
rle	UTAH Run-length Encoded Rasterfiles	read		

icon	Sun Icon Files (old and new format)	read	write	
tiff	Tagged Image File Format	read	write	
face	USENET Face Files		read	write
xbm	X11 Bitmaps		read	write
ilisp	Xerox Interlisp-D Source	read		
brush	Xerox Brush/Press Files		read	
ais	Xerox Array of Intensity Samples	read		

ImageViewer is built around a general image reader/converter/writer that automatically identifies the type of each image by examining the contents of the file. File types (extensions) are only used if the content detector fails. The displayed images can be scaled, resized, and copied to the pasteboard as TIFFs, but that's pretty much all.

To change the size of an image, bring up the inspector panel and type in the new width and height (in pixels). You can also scale it proportionally by entering a single scale factor in the scale field, or two factors (X and Y) separated by a slash. Note that either of these operations will only change your view of the image. If you save it, it will still be in its original size.

This version of ImageViewer also includes a minimal image filtering service that can translate the listed formats to TIFF. The service will automatically be activated when any application that uses the NXImage object and wants to import an image file but doesn't understand the format. Try it by installing ImageViewer in ~/Apps or /LocalApps and drag a non-TIFF image into an Edit window or a Mail message. You should see the icon turn into the real image inside the Edit/Mail window after a short delay. Note that you haven't changed the image file itself, ie. if you send that message, you're really sending the original image file, not a TIFF version of it.

Some nits:

Because ImageViewer nominally can create TIFF files with non-standard bits/sample that currently are unreadable on a NeXT, there is a switch in the preference panel that will avoid this problem by always creating TIFFs with bits/sample values that are powers of two. It is recommended that you always have this turned on, as the difference in saved storage space is minor.

The print function will normally produce output suitable for your chosen printer. However, there are some circumstances under which it may create color PostScript while you only want grayscale. To avoid this, check the preference marked "Print using grayscale". (This used to be more of a problem under 2.X, so you probably won't need to worry about it.)

There is now a default compression choice for TIFF images in the preference panel. Changing this will cause all TIFFs that you save to use the chosen compression. Note that colormapped images can only be saved with without compression; choosing either LZW or JPEG will cause the images to be automatically converted to grayscale or RGB before being saved.

Sorry, alpha (transparency) is still not supported. Images with alpha will receive a white background and lose their transparency.

What about "image"?

"image" is a standalone command line converter based on the same conversion engine as ImageViewer. Please see the supplied man page for instructions on

how to use it.

Changes since the last version (0.9e or so):

- \* Enabled optimized code that converts 8 bit CLUT images to 24 bit RGB. The code was already there, just not used (sigh). This will make ImageViewer decode all 8 bit CLUT images (eg. most GIFs) about 2x faster.
- \* Made ImageViewer recognize and claim .GIF and .JPG files (with capitalized file types).
- \* Added a trivial image filtering service that translates typed filenames to NXTIFFPboardType. Worked around a limitation in services that maximizes the number of send types to 16 by splitting up the declarations in batches.
- \* Fixed bug in code using NXStreams that left image files mapped into memory even after they were closed. This should make ImageViewer much less of a memory hog.
- \* Fixed a bug in the decoding of ASCII PGM images. The checkers are now correctly sized.
- \* Fixed a bug in the code that demultiplies alpha out of 12 bit TIFF images. No more weird golf playing tints.
- \* Added minimal GIF89a support and fixed a bug in the extensions processing code. Mostly the same as before; main difference is that comment fields will be printed on stderr.

- \* Fixed byte ordering dependencies (two if I remember correctly). Fat (m68k, i386, NRW) compiles now work like a charm.
- \* Removed Rainbow support. I don't think anybody uses it anyway.
- \* Added minimal support for Windows BMP file format (does this work?)
- \* Fixed swapped 16 bit input bug in Utah RLE decoder.
- \* Made tiff decoder choose richest representation for multipart tiff files. (Yes, it would be nice to be able to keep them all, but my internal image representation can't handle it. Sorry.)
- \* Fixed bug in grayscale resampling code (im\_togray) that caused some printed images to come out really awful.
- \* Fixed some bugs that caused the right side of printed images to be truncated. Also reduced the margins.
- \* Added a max window size preference that will limit new windows to the given width and height. ShrinkWrapped images will be scaled while ScrollViews will be clipped to conform.
- \* Added a default compression preference to be used when saving TIFF files. Choosing anything else than "None" will cause NeXT's native TIFF routines to be used instead of my own homebrew code.

Disclaimer:

This is unsupported freeware; if it works, it works, if it doesn't, it doesn't. Give it to anyone you like, but please don't charge for it. I don't. Feel free to let me know about any problems, but please don't rely on them being fixed.

ImageViewer is an application that I put together on my spare time while working at Rank Xerox EuroPARC. It is not a product of, nor condoned by NeXT Inc. in any way. Please don't bother them about problems or questions.

(Etc, etc. I think you get the general gist.)

Enjoy,

Lennart Lovstrand  
California, May 1993

lenlo@al.org, Lennart\_Lovstrand@NeXT.COM