

"I swear by my life and my love of it that I will never live for the sake of another man, nor ask another man to live for mine"

John Galt in *Atlas Shrugged*, by Ayn Rand

#### **AUTHOR**

The author is [cristy@mystic.es.dupont.com](mailto:cristy@mystic.es.dupont.com). This software is NOT shareware. However, I am interested in who might be using it. Please consider sending me a picture postcard of the area where you live. Send postcards to

John Cristy  
P.O. Box 40  
Landenberg, PA 19350  
USA

I'm also interested in receiving coins from around the world for my collection.

#### **AVAILABILITY**

ImageMagick is available as

<ftp://ftp.wizards.dupont.com/pub/ImageMagick/ImageMagick-4.1.3.tar.gz>

ImageMagick client executables are available for some platforms. See

<ftp://ftp.wizards.dupont.com/pub/ImageMagick/binaries>  
<ftp://ftp.wizards.dupont.com/pub/ImageMagick/mac>  
<ftp://ftp.wizards.dupont.com/pub/ImageMagick/nt>  
<ftp://ftp.wizards.dupont.com/pub/ImageMagick/vms>  
<ftp://ftp.wizards.dupont.com/pub/ImageMagick/linux>

I want ImageMagick to be of high quality, so if you encounter a problem I will investigate. However, be sure you are using the most recent version from <ftp://ftp.wizards.dupont.com/pub/ImageMagick> before submitting any bug reports or suggestions.

#### **WORLD WIDE WEB**

The official ImageMagick WWW page is

<http://www.wizards.dupont.com/cristy/ImageMagick.html>

To use display as your external image viewer, edit the global mail-cap file or your personal mail-cap file .mailrc (located at your home directory) and put this entry:

```
image/*; display %s
```

#### **MAILING LIST**

There is a mailing list for discussions and bug reports about ImageMagick. To subscribe send the message

```
subscribe magick
```

to majordomo@wizards.dupont.com. You will receive a welcome message which tells you how to post messages to the list, magick@wizards.dupont.com.

## MEMORY REQUIREMENTS

You should allocate sufficient swap space on your system before running ImageMagick; otherwise, you may experience random server or application crashes. Anything less than 80 megabytes of swap space is likely to cause random crashes.

On many systems, you will find that 80 megabytes is insufficient and you will have to allocate more swap space. You should also have at least 32 megabytes of real memory although 64 megabytes or more is recommended.

## UNIX COMPILATION

Type:

```
gunzip ImageMagick-4.1.3.tar.gz
tar xvf ImageMagick-4.1.3.tar
cd ImageMagick
```

If you do not have gunzip(1), it is available as prep.ai.mit.edu:pub/gnu/gzip-1.2.4.shar.

To further enhance the capabilities of ImageMagick, you may wish to install one or more delegate libraries or programs. See MAGICK DELGATES below to learn how to download and build these delegates.

There are currently two mechanisms available to create Makefiles to build ImageMagick: 1) GNU configure, or 2) X11 imake. Each is described in the following paragraphs.

### GNU configure:

This option is easiest to use and is recommended when ImageMagick is to be installed outside of the X11 distribution or working imake configuration files are not available. Use of 'configure' enables automated configuration, building, and installation of PerlMagick.

If you are willing to accept configure's default options, type:

```
./configure
```

and watch the configure script output to verify that it finds everything that you think it should. If it does not, then adjust your environment so that it does.

If you wish to proceed with compiling, go to step 3 below.

If you are not happy with configure's choice of compiler, compilation flags, or libraries, you can give 'configure' initial values for variables by setting them in the environment. Using a Bourne-compatible shell, you can do that on the command line like this:

```
CC=c89 CFLAGS=-O2 LIBS=-lposix ./configure
```

Or on systems that have the 'env' program, you can do it like this:

```
env CPPFLAGS=-I/usr/local/include LDFLAGS=-s ./configure
```

The configure variables you should be aware of are:

CC        Name of C compiler (e.g. 'cc -Xa') to use  
 CFLAGS    Compiler flags (e.g. '-g -O2') to compile with  
 CPPFLAGS   Include paths (-I/somedir) to look for header files  
 LDFLAGS    Library paths (-L/somedir) to look for libraries  
           Systems that support the notion of a library run-path may  
           additionally require -R/somedir or '-rpath /somedir' in order  
           to find shared libraries at run time.  
 LIBS      Extra libraries (-lsomelib) required to link

Any variable (e.g. CPPFLAGS or LDFLAGS) which requires a directory path must specify an absolute path rather than a relative path.

where you left off in By default, 'make install' will install the package's files in '/usr/local/bin', '/usr/local/man', etc. You can specify an installation prefix other than '/usr/local' by giving 'configure' the option '--prefix=PATH'.

Configure can usually find the X include and library files automatically, but if it doesn't, you can use the 'configure' options '--x-includes=DIR' and '--x-libraries=DIR' to specify their locations.

The configure script provides a number of ImageMagick specific options. When disabling an option --disable-something is equivalent to specifying --enable-something=no and --without-something is equivalent to --with-something=no. The configure options are as follows (execute 'configure --help' to see all options).

```
--enable-shared      build shared libraries (default is no)
--enable-static      build static libraries (default is yes)
--enable-lzw         enable LZW support (default is no)
--enable-16bit-pixel enable 16 bit pixels (default is no)
--enable-socks       enable use of SOCKS 5 library and 'rftp'
--with-perl          enable build/install of PerlMagick (default is yes)
--with-bzlib         enable BZlib (default is yes)
--with-dps           enable Display Postscript (default is yes)
--with-fpx           enable FlashPIX (default is yes)
--with-hdf           enable HDF (default is yes)
--with-jbig          enable JBIG (default is yes)
--with-jpeg          enable JPEG (default is yes)
--with-png           enable PNG (default is yes)
--with-tiff          enable TIFF (default is yes)
--with-ttf           enable TrueType (default is yes)
--with-zlib          enable Zlib (default is yes)
--with-x            use the X Window System
```

ImageMagick options represent either features to be enabled or packages to be included in the build. When a feature is enabled (via --enable-something), it enables code already present in ImageMagick. When a package is enabled (via --with-something), the configure script will search for it, and if it is properly installed and ready to use (headers and built libraries are found by compiler) it will be included in the build. The configure script is delivered with all features disabled and all packages enabled (except for PERL). In general, the only reason to disable a package is if a package exists but it is unsuitable for the build (perhaps an old version or not

compiled with the right compilation flags).

Several configure options require special note:

#### **--disable-shared**

the shared libraries are not built. Shared libraries are valuable because they are *\*shared\** across more than one invocation of an ImageMagick or PerlMagick client. In addition, the clients take much less disk space and shared libraries are required in order for PERL to dynamically load the PerlMagick extension.

ImageMagick built with delegates (see MAGICK PLUG-INS below) can pose additional challenges. You can build all the delegates statically and link them into the ImageMagick shared library (i.e. libMagick.so) or alternatively you can build the delegates as shared libraries (some systems already have delegates installed as shared libraries). Shared libraries compilation flags differ from vendor to vendor (gcc's is -fPIC). However, you must compile all shared library source with the same flag (for gcc use -fPIC rather than -fpic).

#### **--disable-static**

static archive libraries (with extension .a) are not built. If you are building shared libraries, there is little value to building static libraries. Reasons to build static libraries include: 1) they can be easier to debug; 2) the clients do not have external dependencies (i.e. libMagick.so); 3) building PIC versions of the delegate libraries may take additional expertise and effort; 4) you are unable to build shared libraries.

**--with-perl** conveniently compile and install PerlMagick in one step. Without this option you first install ImageMagick, change to the PerlMagick subdirectory, build, and finally install PerlMagick. Note, PerlMagick is configured even if --with-perl is disabled. If --enable-shared is not specified, a new PERL interpreter (PerlMagick) is built which is statically linked against the PerlMagick extension. This new interpreter is installed alongside your existing PERL interpreter. If --enable-shared is specified, the PerlMagick extension is built as a dynamically loadable object which is loaded into your current PERL interpreter at run-time. Use of dynamically-loaded extensions is preferable over statically linked extensions so --enable-shared should be specified if possible. If the argument --with-perl=/path/to/perl is supplied, then /path/to/perl will be taken as the PERL interpreter to use.

#### **--with-x=no**

build and use the X11 stubs library (ImageMagick/xlib) instead of the core X11 libraries. This may be necessary on systems where X11 is not installed (e.g. a web server). Display, animate, and import will not work with this library. The remaining programs have reduced functionality.

While configure is designed to ease installation of ImageMagick, it often discovers problems that would otherwise be encountered later when compiling ImageMagick. The configure script tests for headers and libraries by executing the compiler (CC) with the specified compilation flags (CFLAGS), pre-processor flags (CPPFLAGS), and linker flags (LDFLAGS). Any errors are logged to the file 'config.log'. If configure fails to discover a header or library please review this log file to determine why. After taking corrective action, be sure to remove the 'config.cache' file before running configure so that configure will re-inspect the environment rather than using cached values.

Common causes of configure failures are: 1) a delegate header is not in the header include path (CPPFLAGS -I option); 2) a delegate library is not in the linker search/run path (LDFLAGS

-L/-R option); 3) a delegate library is missing a function (old version?); 4) compilation environment is faulty.

If all reasonable corrective actions have been tried and the problem appears to be due to a flaw in the configure script, please send a bug report to the configure script maintainer (currently bfriesen@simple.dallas.tx.us). All bug reports should contain the operating system type (as reported by 'uname -a') and the compiler/compiler-version. A copy of the configure script output and/or the config.log file may be valuable in order to find the problem.

### **X11 Imake:**

Use this option if working imake configuration files are available, the package is to be installed where ever imake installs things (usually the X11 distribution directory), and you don't mind editing a configuration file. Use of this scheme requires a separate step to install PerlMagick (see the README file in the PerlMagick subdirectory).

Review the defines in magick/magick.h and delegates/delegates.mgk and make sure they meet the requirements of your local system.

Edit Magick.tmpl and set the variables to suit your local environment. Now type:

```
xmkmf
make Makefiles
```

or just

```
xmkmf -a
```

if you are using X11R6 imake.

Your build has been configured with either configure or imake.

To build and install ImageMagick, type:

```
make install
```

Note that you may need to be "root" or have special privileges to install ImageMagick in system directories such as /usr/local.

To build and install PerlMagick, type:

```
cd PerlMagick
perl Makefile.PL
make install
```

To confirm your build of the ImageMagick distribution was successful, type:

```
display
```

You may need to type 'rehash' or 'hash -r' if your shell hashes commands, and this is a new install of ImageMagick.

If the program faults, ensure that you have not inadvertently linked to an older version of the libMagick

library. To ensure this is not the case, type:

```
cd ImageMagick/magick
make install
cd ..
make
```

If the image colors are not correct use this command:

```
display -visual default
```

Be sure to read the manual pages for the `display(1)`, `animate(1)`, `montage(1)`, `import(1)`, `mogrify(1)`, `identify(1)`, `combine(1)`, and `convert(1)` utilities. Also read the ImageMagick frequently asked questions in the file `www/Magick.html`. This is required reading. Most of the questions I get via electronic mail are answered in this document.

Place `display(1)` X application defaults in `/usr/lib/X11/app-defaults/Display`. Use the appropriate name for other clients (e.g. `Animate`, `Montage`, etc). To execute `display(1)` from as a menu item of any window manager (`olwm`, `mwm`, `twm`, etc), use

```
logo:Untitled
```

## MAGICK DELEGATES

To further enhance the capabilities of ImageMagick, you may want to get these programs or libraries:

**BZlib** ImageMagick requires the BZLIB library from

<http://www.muraroa.demon.co.uk>

to read and write BZip compressed MIFF images.

**CGM** ImageMagick requires `ralcgm` from

<http://www.agocg.ac.uk/Graphics/CGM/ralcgm.html>

to read the Computer Graphics Metafile image format (may not compile under linux). You also need Ghostscript (see below).

### TransFig

ImageMagick requires `fig2dev` from

[ftp://ftp.x.org/contrib/applications/drawing\\_tools/transfig](ftp://ftp.x.org/contrib/applications/drawing_tools/transfig)

to read the TransFig image format.

### FreeType

ImageMagick requires the FreeType software, version 1.1 or above, available as

<http://www.freetype.org/>

to annotate with TrueType fonts.

**Ghostscript**

ImageMagick requires Ghostscript software available from

<http://www.cs.wisc.edu/~ghost/>

to read the Postscript or the Portable Document format. It is used to annotate an image when an X server is not available. See the FreeType library above for another means to annotate an image. Note, Ghostscript must support the ppmraw device (type `gs -h` to verify). If Ghostscript is unavailable, the Display Postscript extension is used to rasterize a Postscript document (assuming you define HasDPS). The DPS extension is less robust than Ghostscript in that it will only rasterize one page of a multi-page document.

**FlashPix**

ImageMagick requires the FlashPix SDK available from

<http://www.kodak.com/US/en/drg/productsTechnologies/prodTechFlashPix.shtml>

to read and write the FPX image format.

**HDF** ImageMagick requires the NCSA HDF library available via anonymous FTP as

<ftp://ftp.ncsa.uiuc.edu/HDF/HDF4.1r2/tar/HDF4.1r2.tar.gz>

to read and write the HDF image format.

**HP-GL** ImageMagick requires hp2xx available from

<http://www.gnu.org/software/hp2xx/hp2xx.html>

to read the HP-GL image format.

**HTML to Postscript**

ImageMagick requires html2ps available from

<http://www.tdb.uu.se/~jan/html2ps-1.0b1.zip>

to read the HTML image format.

**JBIG** ImageMagick requires the JBIG-Kit software available via anonymous FTP as

<ftp://ftp.informatik.uni-erlangen.de/pub/doc/ISO/JBIG/jbigkit-1.0.tar.gz>

to read the JBIG image format.

**JPEG** ImageMagick requires the Independent JPEG Group's software available via anonymous FTP as

<ftp://ftp.uu.net/graphics/jpeg/jpegsrc.v6b.tar.gz>

to read the JPEG image format. ImageMagick creates progressive JPEG images by default.

Concerning iterative JPEG compression: see Kinoshita and Yamamuro, Journal of Imaging Science and Technology, "Image Quality with Reiterative JPEG Compression", Volume 39, Number 4, July 1995, 306-312 who claim that (1) the iterative factor of the repetitive JPEG operation had no influence on image quality, and (2) the first compression determined base image quality.

**MPEG** ImageMagick requires the MPEG library available via anonymous FTP as

[ftp://ftp.mpeg.org/pub/mpeg/mssg/mpeg2vidcodec\\_v12.tar.gz](ftp://ftp.mpeg.org/pub/mpeg/mssg/mpeg2vidcodec_v12.tar.gz)

to read or write the MPEG image format.

**PNG** ImageMagick requires the PNG library from

<http://www.cdrom.com/pub/png/pngcode.html>

to read the PNG image format.

#### **Radiance**

ImageMagick requires ra\_ppm from Greg Ward's Radiance software available via anonymous FTP as

<http://radsite.lbl.gov/radiance/HOME.html>

to read the Radiance image format (may not compile under linux).

#### **Utah Raster Toolkit**

ImageMagick requires rawtorle from the Utah Raster Toolkit available via anonymous FTP as

<ftp://ftp.cs.utah.edu/pub/urt-3.1b.tar.Z>

to write the RLE image format (may not compile under linux).

#### **Scanner**

ImageMagick requires scanimage from

<http://www.mostang.com/sane/>

to import image from a scanner device.

**TIFF** ImageMagick requires Sam Leffler's TIFF software available via anonymous FTP as

<ftp://ftp.sgi.com/graphics/tiff/tiff-v3.4beta037-tar.gz>

to read the TIFF image format. It in turn optionally requires the JPEG and ZLIB libraries.

#### **SOCKS**

ImageMagick requires SOCKS version 5 available via the Web at

<http://www.socks.nec.com/>



in order for 'xtp' to work across a SOCKS5-based firewall. In particular, 'xtp' makes use of SOCKS5 'rftp' as an external program and supports use of the SOCKS5 library to perform DNS lookups via the firewall rather than the internal DNS server.

### Textures

ImageMagick requires a background texture for the TILE format and for the -texture option of montage(1). You can use your own or get samples from

<http://the-tech.mit.edu/KPT/>

### Web Images

ImageMagick requires GET(1) available via the Web as

<http://www.linpro.no/lwp/>

to read images specified with a World Wide Web (WWW) uniform resource locator (URL). GET(1) must be in /usr/local/bin. See WWWcommand in magick/magick.h to change its location. If you do not have a HTTP server, you can use xtp(1), available in the ImageMagick distribution, for URL's whose protocol is FTP.

**ZLIB** ImageMagick requires the ZLIB library from

<http://www.cdrom.com/pub/infozip/zlib/index.html>

to read or write the PNG or Zip compressed MIFF images.

See the next sections for compiling and installing these delegate libraries and programs.

## HOW TO COMPILE DELEGATE LIBRARIES

NOTE: The following procedure describes how to build ImageMagick delegate libraries in subdirectories of the ImageMagick directory. An alternative to these procedures is to install one or more of these under your system's regular include/lib directory (e.g. the directory specified by --prefix to configure or /usr/local). This allows the libraries to be shared by other packages. When using the configure script, the two schemes may be mixed. Also, please note that when the configure option --enable-shared is not disabled, these procedures must be supplemented with whatever compilation flags are required on your system to generate PIC code. In the case of gcc, this usually means that -fPIC must be added to the compiler options (i.e. CFLAGS) when building each delegate library.

To display images in the HDF, JBIG, JPEG, PNG, TIFF, or TTF format, get the respective archives and build ImageMagick as follows:

### ZLIB

```
cd ImageMagick
gunzip -c bzip2-0.9.0.tar.gz | tar xvof -
mv bzip2-0.9.0 bzlib
cd bzlib
make
```

```
cd ..
```

**HDF**

```
cd ImageMagick
gunzip -c HDF4.1r2.tar.gz | tar xvf -
mv HDF4.1r2 hdf
cd hdf
configure
make -k hdf-libnofortran
cd ..
```

**JBIG**

```
cd ImageMagick
gunzip -c jbigkit-1.0.tar.gz | tar xvof -
mv jbigkit jbig
cd jbig
make
cd ..
```

**JPEG**

```
cd ImageMagick
gunzip -c jpegsrc.v6b.tar.gz | tar xvof -
mv jpeg-6b jpeg
cd jpeg
configure
make
cd ..
```

**PNG**

```
cd ImageMagick
gunzip -c libpng-1.0.2.tgz | tar xvf -
mv libpng-1.0.2 png
cd png
make
cd ..
```

**TIFF**

```
cd ImageMagick
gunzip -c tiff-v3.4beta037.tar.Z | tar xvof -
mv tiff-v3.4beta037 tiff
cd tiff
./configure
make
cd ..
```

**TTF**

```
cd ImageMagick
gunzip -c freetype-1.1.tar.gz | tar xvof -
mv freetype-1.1 ttf
cd ttf
./configure -disable-shared
```

```
make
cd ..
```

**ZLIB**

```
cd ImageMagick
gunzip -c zlib-1.1.3.tar.gz | tar xvf -
mv zlib-1.1.3 zlib
cd zlib
make
cd ..
```

After you have compiled (and optionally installed) the extension libraries you want, continue with where you left off in UNIX COMPILATION.

**VMS COMPILATION**

You might want to check the values of certain program definitions before compiling. Verify the definitions in delegates.mgk to suit your local requirements. Next, type.

Type

```
@make
set display/create/node=node_name::
```

where node\_name is the DECNET X server to contact.

Finally type:

```
display
```

Alternatively, get a zipped distribution (with JPEG, TIFF, PNG, TTF ) from

```
ftp://ftp.wizards.dupont.com/pub/ImageMagick/vms/ImageMagick-4.1.zip
```

The VMS JPEG and TIFF source libraries are available on axp.psl.ku.dk in [anonymous.decwindows.lib].

Thanks to pmoreau@cenaath.cena.dgac.fr for supplying invaluable help as well as the VMS versions of the JPEG, PNG, TIFF, and TTF libraries.

**NT COMPILATION**

The NT distribution contains MetroWerks Codewarrior Professional projects and a Visual C++ workspace (thanks to BillR@corbis.com) for compilation. For those who do not have access to CodeWarrior or Visual C++, the binaries for the command line utilities are enclosed.

If you have an NT X server like Exceed (from Hummingbird) you will also need to include

```
SET DISPLAY=<local-ip-address>:0.0
```

in the System Control panel (NT) or Autoexec.bat (Win95). Autoexec.bat requires that you restart your

computer. See <http://www.rahul.net/kenton/xsites.html> for a list of commercial and free X server software. Without an X server you can still display or animate to, or import from, a remote X server. Convert, mogrify, montage, combine, and identify will work with or without an X server directly from the command prompt.

To view any image in a Microsoft window, type

```
convert image.ext win:
```

Import(1) works if you have at least one X window open. Alternatively, type

```
convert x:root image.gif
```

Make sure gswin32 (Ghostscript) is in your execution path (see Autoexec.bat), otherwise, you will be unable to convert or view a Postscript document.

Make sure iexplore (Internet Explorer) is in your execution path (see Autoexec.bat), otherwise, you will be unable to browse the ImageMagick documentation.

To compile the source with Codewarrior, start with Magick/Magick.mcp and then animate.mcp, convert.mcp, etc.. The Visual C++ workspace is ImageMagick.dsw.

And yes, the NT executables will work under Windows 95.

## MACINTOSH COMPILATION

The Macintosh distribution contains MetroWerks Codewarrior Professional projects for compilation. For those who do not have access to CodeWarrior, the binaries for the command line utilities are enclosed. I had to comment the inline intrinsic functions in math.h in order to compile. If you have a better solution, let me know.

Display(1), animate(1), and import(1) currently do not work on the Macintosh.

I am looking for a volunteer to get display(1) and animate(1) to work on the Macintosh. I also need a volunteer is needed to write a simple Mac program to call the libMagick routines and display an image in a window.

## ANIMATION

To prevent color flashing on visuals that have colormaps, animate(1) creates a single colormap from the image sequence. This can be rather time consuming. You can speed this operation up by reducing the colors in the image before you 'animate' them. Use mogrify(1) to color reduce the images:

```
mogrify +map -colors 256 scenes/dna.[0-9]*
```

Alternatively, you can use a Standard Colormap; or a static, direct, or true color visual. You can define a Standard Colormap with xstcmap(1). For example, to use the "best" Standard Colormap, type:

```
xstcmap -best  
animate -map best scenes/dna.[0-9]*
```

or to use a true color visual:

```
animate -visual truecolor scenes/dna.[0-9]*
```

Image filenames can appear in any order on the command line if the scene keyword is specified in the MIFF image. Otherwise the images display in the order they appear on the command line. A scene is specified when converting from another image format to MIFF by using the "scene" option with any filter. Be sure to choose a scene number other than zero. For example, to convert a TIFF image to a MIFF image as scene #2, type:

```
convert -scene 2 image.tiff image.miff
```

## 16-BIT IMAGING

By default, ImageMagick uses a color depth of 8 bits (e.g. [0..255] for each of red, green, blue, and transparency components). Any 16-bit image is scaled to 8-bits before any image viewing or processing occurs. If you want to work directly with 16-bit images (e.g. [0..65535]), edit Magick.tmpl and define QuantumLeap or use -enable-16bit with configure. Next, type:

```
make clean
make
```

In 16-bit mode expect to use about 33% more memory on the average. Also expect some processing to be slower than in 8-bit mode (e.g. Oil Painting, Segment, etc).

In general, 16-bit mode is only useful if you have 16-bit images that you want to manipulate and save the transformed image back to a 16-bit image format (e.g. PNG, VIFF).

## 64-BIT MACHINES

Each pixel, within ImageMagick, is represented by the RunlengthPacket structure found in magick/image.h. Only 8 bits are required for each color component and 16 bits for the colormap index for a total of 6 bytes. If QuantumLeap is defined (see 16-BIT IMAGING above), the color component size increases to 16 bits for a total of 10 bytes. Some 64-bit machines pad the structure which can cause a significant waste of memory. For the cray, change the RunlengthPacket structure to this

```
typedef struct _RunlengthPacket
{
    unsigned char
        red : QuantumDepth,
        green : QuantumDepth,
        blue : QuantumDepth,
        length : QuantumDepth;

    unsigned short
        index : 16;
} RunlengthPacket;
```

before compiling.

I'm not sure if this will work on other 64-bit machines that pad. If you know a better solution, please send me E-mail. Note, that the Dec Alpha apparently does not pad the structure so ImageMagick should be fine on this particular 64-bit machine.

**ACKNOWLEDGEMENT**

ImageMagick has been significantly enhanced due to the efforts of these individuals:

Bob Friesenhahn <bfriesen@simple.dallas.tx.us>: Configure scripts  
and the initial design of the ImageMagick delegate subsystem.

Bill Radcliffe <BillR@corbis.com>: FlashPix support

And thanks to the hundreds of people that have sent e-mail with bug reports and suggestions for improving ImageMagick.

**COPYRIGHT**

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files ("ImageMagick"), to deal in ImageMagick without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of ImageMagick, and to permit persons to whom the ImageMagick is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of ImageMagick.

The software is provided "as is", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and noninfringement. In no event shall E. I. du Pont de Nemours and Company be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with ImageMagick or the use or other dealings in ImageMagick.

Except as contained in this notice, the name of the E. I. du Pont de Nemours and Company shall not be used in advertising or otherwise to promote the sale, use or other dealings in ImageMagick without prior written authorization from the E. I. du Pont de Nemours and Company.