NAME

imconv - Convert between image file formats

SYNOPSIS

imconv [*options*] *infilename outfilename*

DESCRIPTION

With no recognized standard, most commercially available image-handling applications use their own custom image file format to store pixel data. To use imagery generated by one application as input into another is complicated by the need to convert from one custom file format to another.

The SDSC image tools are tools developed at the San Diego Supercomputer Center (SDSC) to handle image manipulation and file format conversion for a wide range of file formats.

imconv converts an input image file with one format to an output image file with a different format. If the input file contains multiple images, so will the output file, if its format can support it.

OPTIONS

imconv has a variety of options in the following four categories:

File Selection	What input and output files to use
Format Selection	What image file format to use
Format Control	What variant of a file format to generate
Standard	Standard generic options on all SDSC tools

These categories are discussed separately below.

All options can be abbreviated to the first few unique characters.

File Selection Options

imconv needs to know where to read image information and where to write it. So, in its simplest form, it requires only the input and output filenames as follows:

```
imconv picture.pix newpic.ras
```

imconv assumes the first filename is the input file and the second the output file.

You may optionally precede the input and output filenames by **-infile**, and **-outfile**, respectively. For instance:

```
imconv -infile picture.pix -outfile newpic.ras
```

To direct **imconv** to read from **stdin** or write to **stdout**, use a filename consisting of a single dash (-).

```
imconv -pix - newpic.ras < picture.pix
   or
imconv picture.pix -ras - > newpic.ras
   or
imconv -pix - -ras - < picture.pix > newpic.ras
   or
cat picture.pix | imconv -pix - -ras - | cat > newpic.ras
```

When reading from **stdin** or writing to **stdout**, you must indicate the format of the image file explicitly with options like **-pix** and **-ras** above. These options are discussed in more detail in the next section.

Format Selection Options

imconv supports the following image file formats:

Format Names		
Primary	**Others**	**Description**
gif	giff	CompuServe Graphics Image Format File
hdf	df, ncsa	Hierarchical Data Format file
icon	cursor, pr	Sun Icon and Cursor file
iff	suniff, taac	Sun TAAC Image File Format
mpnt	macp, pntg	Apple Macintosh MacPaint file
pbm	-	Portable Bit Map file
pcx	pcc	ZSoft PC Paintbrush file
pgm	-	Portable Grayscale Map file
pic	picio, pixar	PIXAR PICture file
pict	pict2	Apple Macintosh QuickDraw/PICT picture file
pix	alias	Alias PIXel image file
pnm	-	Portable aNy Map file
ppm	-	Portable Pixel Map file
ps	postscript	PostScript image file
ras	sun, sr, scr	Sun RASterfile
rgb	iris, sgi	Silicon Graphics RGB image file
rla	rlb	Wavefront raster image file
rle	-	Utah Run-Length-Encoded image file
rpbm	-	Raw Portable Bit Map file
rpgm	-	Raw Portable Grayscale Map file
rpnm	-	Raw Portable aNy Map file
rppm	-	Raw Portable Pixel Map file
synu	-	Synu image file
tiff	tif	Tagged Image File
x	avs	Stardent AVS X image file
xbm	bm	X11 Bit Map file
xwd	x11	X11 Window Dump image file

For **imconv** to read or write an image file, it must know which of the above formats it is reading or writing. In most cases, **imconv** can determine your intent by looking at the input and output filenames. For instance:

```
imconv picture.pix mypic.ras
```

To determine the format of **picture.pix**, **imconv** opens it and looks at the first few bytes of the file. It checks these against a list of "magic numbers" for various image file formats. If it finds no match, **imconv** extracts the filename suffix (**.pix** from **picture.pix**) and compares it against a list of known suffixes for supported image file formats.

To determine the format to use for **mypic.ras**, **imconv** extracts the output filename suffix (**.ras** from **mypic.ras**) and compares it against a list of suffixes for supported image file formats.

If **imconv** cannot discern what format to use, it issues an error message and exits.

On occasion it is necessary to override **imconv**'s file format assumptions. To do so, you must specify an explicit format option preceding an input or output filename. For instance:

```
imconv -pix picture.pix -ras mypic.ras
```

This is required when reading from **stdin** or writing to **stdout**. In either case **imconv** has no filename from which to extract a filename suffix and, therefore, cannot discern what format to use.

The first column in the format table above gives the most commonly used filename suffix for each of the supported image file formats. Because multiple filename suffixes are in common use for a given file format, **imconv** recognizes a variety of names for each format. These additional names are given in the second column of the table. You can obtain a complete list of the image format names (filename suffixes and format options) by entering **imconv** with the **-fullhelp** option, as follows:

```
imconv -fullhelp
```

You can use any of these names, preceded by a dash, to specify the input and output formats. If used, the format selection option must precede the filename to which it refers.

You can obtain more information on the file formats by using **imformats**(1IM) with the **-long** option, as follows:

```
imformats -long
```

Format Control Options

Virtually all image file formats have multiple variants. These variants have different image depths, different compression schemes, different RGB image interleave methods, inclusion or exclusion of color lookup tables (CLTs) and alpha planes, and so on. In most cases **imconv** picks the correct variant to use when writing the output file. However, in some cases you may wish to override **imconv**'s defaults and select a variant explicitly using the following format control options:

-outindex
-outrgb

Output to color index or RGB image, respectively.

Typically, image pixels are represented as color index (pseudo-color) or RGB (true-color) values.

A color index is a small integer (usually 8- or 12-bits) that indexes into an associated color lookup table (CLT) to get the red-green-blue (RGB) color value for a pixel in the image. A color index image is often refered to as a "pseudo-color" image because the color for a given pixel is found in the CLT, not in the pixel data itself.

An RGB image stores the red-green-blue (RGB) value for a pixel's color for every pixel in the image. There is no associated CLT. RGB images are often refered to as "true-color" because the full color description (RGB value) is stored in each pixel.

-outindex forces **imconv** to convert the incoming image to a color index image before writing it to the output file. If the output file format cannot support storing a color index image, then **imconv** reports an error and exits.

-outrgb forces **imconv** to convert the incoming image to an RGB image before writing it to the output file. If the output file format cannot support storing an RGB image, then **imconv** reports an error and exits.

By default, if the input file's image is a color index image, **imconv** tries to write it out as a color index image. Similarly, if the input file's image is an RGB image, **imconv** tries to write it out as an RGB image. If the output format doesn't support what **imconv** wants, **imconv** automatically converts the image to one of the supported output file format variants.

-outchandepth *nbits*

Specify output depth.

Some file formats allow color indexes and RGB values to be stored with a specific number of "bits-per-channel". While 8 bits-per-channel is by far the most common, some formats support 1, 4, 12, 16, 24, and 32 bits-per-channel.

A color index image has 1 channel (the index into the color lookup table); an RGB image has 3 channels (red, green, and blue). The channel depth is the number of bits *for each channel*. So, an RGB image with an 8-bit channel depth takes 3 ∗ 8 = 24 bits per pixel. An RGB image with a 24-bit channel depth takes 3 ∗ 24 = 72 bits per pixel.

The *nbits* argument to the **-outchandepth** is the number of bits to use per channel when writing out the image. If the output file format cannot support the number of bits per channel you specify, **imconv** reports an error and exits.

By default, **imconv** chooses the best number based on the number of bits per channel used by the incoming image. If an input file's image uses a 16-bit color index, **imconv** tries to output the image using a 16-bit, or more, color index. Truncation occurs only if the output format cannot support the incoming image's number of bits per channel, or anything larger. In fact, in the case of color index images, if **imconv** is given the choice of truncating a 16-bit color index image to 8 bits or converting it to RGB for the same output format, **imconv** converts it to RGB. **imconv** always chooses the path with the least degradation of the data.

-outnchan *nchan*

Select number of output channels.

The number of channels in an image is the number of values stored per pixel. A color index image has 1 channel (the color index into the color lookup table). An RGB image has 3 channels (red, green, and blue). Though channel numbers other than 1 or 3 are possible, in practice they aren't used. (We don't count the alpha channel.)

The *nchan* argument to the **-outnchan** option selects the number of channels per pixel (not including the alpha channel) to use when writing the image.

This option is provided as a path towards future functionality. In the present release, this option provides the same information as the **-outindex** and **-outrgb** options.

-outclt
-outnoclt

Do or don't output a color lookup table (CLT).

Color index images usually have a CLT associated with them. RGB images can also have them, though less commonly.

Some image file formats allow the CLT to be left out of the image file. The **-outnoclt** option directs **imconv** to not write the CLT. The **-outclt** option directs **imconv** to write the CLT.

Some formats require the CLT to be included in the file. In such cases, if you specify the **-outnoclt** option, **imconv** reports an error and exits, since it is unable to comply with your request.

In much rarer cases, some formats do not allow a CLT to be stored in the image file. If you specify the **-outclt** option, **imconv** reports an error and exits.

If the incoming image does not have a CLT but you specify **-outclt**, **imconv** writes a ramping CLT (low to high). This is most common when reading in a grayscale image (color index image without a CLT) and writing out a color image (color index image with a gray ramp for a CLT).

By default, if the incoming image has a CLT, **imconv** attempts to store it. If the output format cannot store a CLT but can store an RGB image, **imconv** automatically converts a color index image to an RGB image and stores it that way.

-outalpha
-outnoalpha

Do or don't output an alpha channel.

An alpha channel contains a coverage value for each image pixel and typically is used when compositing images. For instance, to make image ABC partially cover XYZ, parts of ABC are marked as opaque (they cover XYZ), while other parts are transparent (XYZ shows through).

Alpha values typically range from 0 to 255. 0 means transparent, 255 opaque. Values between 0 and 255 indicate partial transparency and direct compositing software to mix the color of the image on top with the color of the image below, using the alpha value as a weighting factor.

-outalpha directs that an alpha channel be output for the image. If **-outalpha** is given and the output format cannot support an alpha channel (most cannot), **imconv** reports an error and exits.

-outnoalpha directs that an alpha channel not be output. If **-outnoalpha** is given and the output format must have an alpha channel (rare), **imconv** reports an error and exits.

If the incoming image does not have an alpha channel but you specify **-outalpha**, **imconv** automatically generates an opaque alpha channel (all 255s).

By default, if the incoming image has an alpha channel, **imconv** tries to write it out. For some file formats, only RGB images, not color index images, can be stored with alpha channels. In such cases, if the incoming image is a color index image with an alpha channel, **imconv** automatically converts the image to RGB to preserve the alpha channel.

-outinterleave *method*

Specify interleave method.

This option only applies to RGB images.

RGB images are written out using one of the following three methods:

noninterleaved	RGBRGBRGBRGB...
scanline-interleaved	RR..GG..BB..RR..GG..BB...

plane-interleaved	RRRR..GGGG..BBB..

Some formats support all three methods. Other formats support only one or two. The **-outinterleave** option takes one of the following arguments:

none	noninterleaved
line	scanline-interleaved
plane	plane-interleaved

If the output format does not support the interleave method you select, **imconv** reports an error and exits.

By default, **imconv** chooses the most efficient or most widely used interleave method for the output format. In most cases **imconv** chooses plane-interleaved over scanline-interleaved, and scanline-interleaved over noninterleaved. This is because image compression schemes (see below) work better on plane- and scanline-interleaved images, thus reducing the amount of disk space an image file requires.

-outcompress *scheme*

Specify compression scheme.

Most image file formats support compression schemes to reduce the size of a file. Some formats even support multiple compression schemes.

The single argument to the **-outcompress** option is the name of the compression scheme to use:

none	noncompressed
lzw	Limpel-Ziv & Welsh compressed
pb, packbits, mac	Apple Macintosh Packbits
rle	Run-Length-Encoded

Noncompressed images store each image pixel as a value or values in the file. No tricks are applied to reduce the disk space requirements. Noncompressed image files take longer to write out and read in, and they use up more disk space (often several times as much as a wellcompressed image).

Compression schemes use trickery to reduce the storage requirements of the image. The most common among these is Run-Length Encoding (RLE). RLE is based on the fact that most images have runs of adjacent pixels of the same color, such as solid color image backgrounds. These runs can be abbreviated to just a count of the number of pixels in the run, and the color to use for the run. Runlength-encoded images usually take 30-50% less disk space.

Most image formats support some variant of RLE compression. Since these variants are similar, **imconv** refers to them all as "rle compression."

Limpel-Ziv & Welsh and Apple's Packbits compression are fancier and too complex to explain here.

If the output format does not support the form of compression you select with the **-outcompress** option, **imconv** reports an error and exits.

By default, **imconv** chooses the most efficient or most widely used compression scheme supported by the output file format.

Most image file formats only support a subset of the functionality represented by **imconv**'s options. In rare cases, the format may support the functionality, but **imconv** does not. For a list of what is supported, enter **imformats**(1IM) with the **-long** option, as follows:

```
imformats -long
```

For greater detail, you may specify the **-long** option twice, as follows:

```
imformats -long -long
```

Standard Options

imconv recognizes the following standard SDSC options:

-feedback
Create a software feedback (bug report) form in the file **imconv.fbk.0**.

-fullhelp
Display a detailed list of the arguments and how to use **imconv**.

-help
Display an abbreviated list of the arguments and how to use **imconv**.

-register
Create a software user registration form in the file **imconv.reg.0**.

-verbose
Display progress messages to **stdout** during the course of the image file conversion.

-version
Display the version number and copyright for **imconv**.

NOTES

Error messages are reported to **stderr**.

The conversions take varying lengths of time depending upon the complexity of the input and output file formats and the complexity of the image itself.

Some file formats create a temporary file in **/usr/tmp** when reading from **stdin** or writing to **stdout**.

Some file formats, such as PostScript, cannot be used for input.

Typically, you don't need to use the various channel depth, interleave method, and compression scheme options. **imconv** does a good job of figuring out how to get the input file's image stored into the output file with a minimum loss of information, if any at all. If anything, **imconv** is overly zealous about avoiding loss of information. Consider this scenario: An input file contains a color index image with a CLT. The selected output format supports both color index (with or without CLT) and RGB images. The user enters the following command line:

```
imconv input.ras -outnoclt output.hdf
```

What type of image is stored in **output.hdf**? The answer: An RGB image, even though the incoming image was a color index image!

The intent of the user typing in the above command might be to strip off the CLT and write the color index image to the output file. However, **imconv** interprets the **-outnoclt** option literally. **-outnoclt** means a CLT should not be output. But it gives no information about how image pixels should be stored. To avoid loss of information yet satisfy the user's request, **imconv**'s only recourse is to convert the image to RGB.

To strip off the CLT but preserve the image as a color index image, enter the following command line instead:

```
imconv input.ras -outindex -outnoclt output.hdf
```

EXAMPLES

To convert an Alias **pix** file to an **hdf** file, enter any of the following command lines:

```
imconv picture.pix picture.hdf
   or
imconv -pix picture.pix -hdf picture.hdf
   or
imconv -pix - picture.hdf < picture.pix
   or
imconv -pix picture.pix -hdf - > picture.hdf
   or
imconv -pix - -hdf - < picture.pix > picture.hdf
   or
cat picture.pix | imconv -pix - -hdf - | cat > picture.hdf
```

To convert an RGB image stored in a **pix** format file to an RGB image stored in a **ras** format file, enter the following:

```
imconv truecolor.pix truecolor.ras
```

To convert the same RGB **pix** file image to an 8-bit-per-channel color index image in a **ras** file, enter any of the following:

```
imconv truecolor.pix -outindex -outchandepth 8 -outnchan 1 pseudocolor.ras
   or
imconv truecolor.pix -outindex -outchandepth 8 pseudocolor.ras
   or
imconv truecolor.pix -outindex pseudocolor.ras
```

To select Apple Macintosh Packbits compression and plane-interleaved RGB storage for a **tiff** file, enter any of the following:

```
imconv picture.pix -outcomp pb -outinter plane picture.tiff
   or
imconv picture.pix -outcomp packbits -outinter plane picture.tiff
   or
imconv picture.pix -outcomp mac -outinter plane picture.tiff
```

To read in an uncompressed Sun Rasterfile and compress it into a new file, enter the following:

```
imconv uncompressed.ras -outcomp rle compressed.ras
```

To convert a Macintosh **pict** file drawing into an X Window System Bit Map for use as an icon, cursor, or whatever, enter the following:

```
imconv icon.pict icon.xbm
```

SEE ALSO

imcopy(1IM), **imfile**(1IM), **imflip**(1IM), **imformats**(1IM), **imgray**(1IM), **immono**(1IM), **impaste**(1IM), **imscale**(1IM)

For information on SDSC's image library, see **imintro**(3IM).

For information on the individual image file formats, see their respective **man** pages: **imeps**(3IM), **imgif**(3IM), **imhdf**(3IM), **imicon**(3IM), **imiff**(3IM), **immpnt**(3IM), **impbm**(3IM), **impcx**(3IM), **impgm**(3IM), **impic**(3IM), **impict**(3IM), **impix**(3IM), **impnm**(3IM), **imppm**(3IM), **imps**(3IM), **imras**(3IM), **imrgb**(3IM), **imrla**(3IM), **imrle**(3IM), **imrpbm**(3IM), **imrpgm**(3IM), **imrpnm**(3IM), **imrppm**(3IM), **imsynu**(3IM), **imtiff**(3IM), **imx**(3IM), **imxbm**(3IM), **imxwd**(3IM).

KNOWN PROBLEMS

Different formats use different terminology. **imconv** uses generic terminology. If you're only familiar with format-specific terms, you may find **imconv** confusing.

imconv supports selection of only the most common format-specific variants. For instance, you can't select how GIF 8-bit color index images should be interlaced when stored (not the same as RGB interleaving).

When an input file has multiple images in it (such as HDF, GIF, or TIFF files), the output file format must also support multiple images per file. SDSC plans to implement tools in the near future to create and split multi-image files.

AUTHOR

Dave Nadeau
San Diego Supercomputer Center

See the individual file format **man** pages for the authors of the underlying format read and write code. The names of these **man** pages begin with the letters "im" followed by the format name. For example, the name of the TIFF **man** page is **imtiff**. To display it, enter **man imtiff**.

CONTACT

SDSC consultants, (619)534-5100, **consult@y1.sdsc.edu**

Notes

NAME

imcopy - Copy a portion of an image to a new file

SYNOPSIS

imcopy [*options*] *infilename outfilename*

DESCRIPTION

imcopy copies a portion of an input image to a new file. If the input file contains multiple images, a portion of each input image is copied and written to the output file. The input and output image file formats may be different.

OPTIONS

imcopy has a variety of options in the following five categories:

File Selection	What input and output files to use
Format Selection	What image file format to use
Format Control	What variant of a file format to generate
Standard	Standard generic options on all SDSC tools
Manipulation	How the image can be manipulated

File Selection, *Format Selection*, *Format Control*, and *Standard* options are common to all SDSC image tools and are discussed in depth in the **man** page for **imconv**(1IM).

All options can be abbreviated to the first few unique characters.

Manipulation Options

The region of the input image to be copied to a new file is defined by four arguments:

-xposition *x*	Left edge of region
-yposition *y*	Top edge of region
-xsize *w*	Width of region
-ysize *h*	Height of region

-xposition and **-yposition** specify the copy region position and are constrained to be within the bounds of the input image. (0,0) is the upper left corner of the image. If these options are not given, the copy region position defaults to a position that centers the region within the input image.

-xsize and **-ysize** give the size of the copy region and are constrained to fit within the bounds of the input image. If these options are not given, the copy region defaults to a rectangle whose upper left corner is at the given copy region position, and which extends to the right and bottom edges of the input image.

NOTES

For notes regarding file format conversion and standard image tool options, see the **man** page on **imconv**(1IM).

Error messages are reported to **stderr**.

EXAMPLES

To copy from an Alias PIX file a 100x100 pixel region whose upper left corner is at (10,20) and store it to a Wavefront RLA file, use the following:

```
imcopy picture.pix -xpos 10 -ypos 20 -xsize 100 -ysize 100 clipart.rla
```

To copy a 500x500 pixel region from the center of an HDF file and write it to a TIFF file, use the following:

```
imcopy picture.hdf -xsize 500 -ysize 500 newpic.tiff
```

To copy the top 20 scanlines from an XWD file and store it to a new XWD file, use the following:

```
imcopy window.xwd -ypos 0 -ysize 20 border.xwd
```

To copy an entire input GIF image to a PCX file, use the following:

```
imcopy incoming.gif outgoing.pcx
```

With no copy region specified, the actions of **imcopy** default to the same as the SDSC image format conversion tool **imconv**(1IM), but they take a little longer.

SEE ALSO

imconv(1IM), **imfile**(1IM), **imflip**(1IM), **imformats**(1IM), **imgray**(1IM), **immono**(1IM), **impaste**(1IM), **imscale**(1IM)

For information on SDSC's image library, see **imintro**(3IM).

For information on the individual image file formats, see their respective **man** pages: **imeps**(3IM), **imgif**(3IM), **imhdf**(3IM), **imicon**(3IM), **imiff**(3IM), **immpnt**(3IM), **impbm**(3IM), **impcx**(3IM), **impgm**(3IM), **impic**(3IM), **impict**(3IM), **impix**(3IM), **impnm**(3IM), **imppm**(3IM), **imps**(3IM), **imras**(3IM), **imrgb**(3IM), **imrla**(3IM), **imrle**(3IM), **imrpbm**(3IM), **imrpgm**(3IM), **imrpnm**(3IM), **imrppm**(3IM), **imsynu**(3IM), **imtiff**(3IM), **imx**(3IM), **imxbm**(3IM), **imxwd**(3IM).

AUTHOR

Dave Nadeau
San Diego Supercomputer Center

See the individual file format **man** pages for the authors of the underlying format read and write code. The names of these **man** pages begin with the letters "im" followed by the format name. For example, the name of the TIFF **man** page is **imtiff**. To display it, enter **man imtiff**.

CONTACT

SDSC consultants, (619)534-5100, **consult@y1.sdsc.edu**

NAME

imfile - Discern the image format of a file(s)

SYNOPSIS

imfile [*options*] *filename1 filename2 ...*

DESCRIPTION

imfile attempts to discern the image format of each file listed on the command line, much like the UNIX **file**(1) utility. **imfile** outputs the name of the file and its image file format to **stdout**.

OPTIONS

Invoke **imfile** with one or more filenames, optionally preceded by the **-infile** argument:

```
imfile -infile picture.pix newpic.ras
   or
imfile picture.pix newpic.ras
```

imfile also recognizes the following standard SDSC options:

-feedback
Create a software feedback (bug report) form in the file **imfile.fbk.0**.

-fullhelp
Display a detailed list of the arguments and how to use **imfile**.

-help
Display an abbreviated list of the arguments and how to use **imfile**.

-register
Create a software user registration form in the file **imfile.reg.0**.

-version
Display the version number and copyright for **imfile**.

All options can be abbreviated to the first few unique characters.

NOTES

imfile discerns the type of a file first by checking the file's magic number against a magic number list for known image formats. If the magic number doesn't match any of those in the list, **imfile** looks at the filename's suffix (the characters following the last period in the filename) and checks it against a list of known image filename suffixes.

If **imfile** cannot determine the format of a file, it outputs the string "Unknown image file format."

EXAMPLES

Discern the types of three files:

```
imfile thing1.pix thing2.hdf dr.who
```

The above command outputs:

```
thing1.pix:    'pix', Alias image file, Alias Research, Inc.
thing2.hdf:    'hdf', Hierarchical Data File, NCSA
dr.who:        Unknown image file format
```

SEE ALSO

imconv(1IM), **imcopy**(1IM), **imflip**(1IM), **imformats**(1IM), **imgray**(1IM), **immono**(1IM), **impaste**(1IM), **imscale**(1IM)

For information on SDSC's image library, see **imintro**(3IM).

For information on the individual image file formats, see their respective **man** pages: **imeps**(3IM), **imgif**(3IM), **imhdf**(3IM), **imicon**(3IM), **imiff**(3IM), **immpnt**(3IM), **impbm**(3IM), **impcx**(3IM), **impgm**(3IM), **impic**(3IM), **impict**(3IM), **impix**(3IM), **impnm**(3IM), **imppm**(3IM), **imps**(3IM), **imras**(3IM), **imrgb**(3IM), **imrla**(3IM), **imrle**(3IM), **imrpbm**(3IM), **imrpgm**(3IM), **imrpnm**(3IM), **imrppm**(3IM), **imsynu**(3IM), **imtiff**(3IM), **imx**(3IM), **imxbm**(3IM), **imxwd**(3IM).

AUTHOR

Dave Nadeau
San Diego Supercomputer Center

See the individual file format **man** pages for the authors of the underlying format read and write code. The names of these **man** pages begin with the letters "im" followed by the format name. For example, the name of the TIFF **man** page is **imtiff**. To display it, enter **man imtiff**.

CONTACT

SDSC consultants, (619)534-5100, **consult@y1.sdsc.edu**

NAME

imflip - Flip images vertically or horizontally and store in a new file

SYNOPSIS

imflip [*options*] *infilename outfilename*

DESCRIPTION

imflip reads each image in the input file and flips it vertically, horizontally, or both, then writes it to the output file. The input and output image file formats may be different.

OPTIONS

imflip has a variety of options in the following five categories:

File Selection	What input and output files to use
Format Selection	What image file format to use
Format Control	What variant of a file format to generate
Standard	Standard generic options on all SDSC tools
Manipulation	How the image can be manipulated

File Selection, Format Selection, *Format Control*, and *Standard* options are common to all SDSC image tools and are discussed in depth in the **man** page for **imconv**(1IM).

All options can be abbreviated to the first few unique characters.

Manipulation Options

-xflip flips the image left-to-right.

-yflip flips the image top-to-bottom.

-xflip and **-yflip** together flip the image left-to-right and top-to-bottom.

NOTES

For notes regarding file format conversion and standard image tool options, see the **man** page on **imconv**(1IM).

Error messages are reported to **stderr**.

EXAMPLES

To flip a Wavefront RLB file top-to-bottom and store the result into an SGI RGB file, use the following:

```
imflip rightsideup.rlb -yflip upsidedown.rgb
```

To flip a TIFF image left-to-right and top-to-bottom, use the following:

```
imflip unflipped.tiff -xflip -yflip flipped.tiff
```

To copy an entire input GIF image to a PCX file without any flipping, use the following:

```
imflip incoming.gif outgoing.pcx
```

With no flip arguments, the actions of **imflip** default to the same as the SDSC image format conversion tool **imconv**(1IM), but they take a little longer.

SEE ALSO

imconv(1IM), **imcopy**(1IM), **imfile**(1IM), **imformats**(1IM), **imgray**(1IM), **immono**(1IM), **impaste**(1IM), **imscale**(1IM)

For information on SDSC's image library, see **imintro**(3IM).

For information on the individual image file formats, see their respective **man** pages: **imeps**(3IM), **imgif**(3IM), **imhdf**(3IM), **imicon**(3IM), **imiff**(3IM), **immpnt**(3IM), **impbm**(3IM), **impcx**(3IM),

impgm(3IM), **impic**(3IM), **impict**(3IM), **impix**(3IM), **impnm**(3IM), **imppm**(3IM), **imps**(3IM), **imras**(3IM), **imrgb**(3IM), **imrla**(3IM), **imrle**(3IM), **imrpbm**(3IM), **imrpgm**(3IM), **imrpnm**(3IM), **imrppm**(3IM), **imsynu**(3IM), **imtiff**(3IM), **imx**(3IM), **imxbm**(3IM), **imxwd**(3IM).

AUTHOR

Jim McLeod
San Diego Supercomputer Center

See the individual file format **man** pages for the authors of the underlying format read and write code. The names of these **man** pages begin with the letters "im" followed by the format name. For example, the name of the TIFF **man** page is **imtiff**. To display it, enter **man imtiff**.

CONTACT

SDSC consultants, (619)534-5100, **consult@y1.sdsc.edu**

NAME

imformats - List information on image file formats

SYNOPSIS

imformats [*options*]

DESCRIPTION

imformats displays a list of the image file formats supported by the SDSC image tools (such as **imconv**(1IM)) and the underlying SDSC image library **libim.a** (see **imintro**(3IM)).

OPTIONS

With no options given, **imformats** prints to **stdout** a list of supported image file formats. For instance:

```
imformats
```

The above command displays:

```
Format   Description
-------  ---------------------------------------------
gif      CompuServe Graphics Image Format File
hdf      Hierarchical Data Format File
icon     Sun Icon and Cursor file
iff      Sun TAAC Image File Format
mpnt     Apple Macintosh MacPaint file
pbm      Portable Bit Map file
pcx      ZSoft IBM PC Paintbrush file
pgm      Portable Grayscale Map file
pic      PIXAR PICture file
pict     Apple Macintosh QuickDraw/PICT file
pix      Alias PIXel image file
pnm      Portable aNy Map file
ppm      Portable Pixel Map file
ps       PostScript image file
ras      Sun RASterfile
rgb      Silicon Graphics RGB image file
rla      Wavefront raster image file
rle      Utah Runlength-encoded image file
rpbm     Raw Portable Bit Map file
rpgm     Raw Portable Grayscale Map file
rpnm     Raw Portable aNy Map file
rppm     Raw Portable Pixel Map file
synu     Synu image file
tiff     Tagged Image File
x        Stardent AVS X image file
xbm      X11 Bit Map file
xwd      X Window Dump image file
```

To restrict the table to one or more formats, list the names of the formats on the command line, as in the following:

```
imformats -ras -tiff -hdf
```

which displays:

```
Format    Description
-------   ----------------------------------
ras       Sun Rasterfile
tiff      Tagged Image File Format
hdf       Hierarchical Data File
```

Information on file formats is displayed in the order you specify on the command line.

The **-long** option elaborates on the list by including information on the format’s creator or vendor, other common names for the format, and a summary of the supported format variants for read and write operations. For instance, to find out more about support for TIFF files, enter the following:

```
imformats -tiff -long
```

which displays:

```
tiff   Tagged image file
         a.k.a.:   tif
         Creator:   Aldus, Microsoft, and NeXT
         Read support:
            1-, 4-, 8-, and 32-bit color index images.  24-bit RGB and 32-bit RGB+alpha images.
            Standard (uncompressed), Mac Packbits, and Lempel-Ziv & Welsh compression.
         Write support:
            1-, 8-, and 32-bit color index images.  24-bit RGB and 32-bit RGB+alpha images.
            Standard (uncompressed), Mac Packbits, and Lempel-Ziv & Welsh compression.
```

If you specify **-long** twice, you receive a detailed list of the variants of the file format that can be read and written. For instance:

```
imformats -tiff -long -long
```

displays:

```
tiff   Tagged image file
         a.k.a.:   tif
         Creator:   Aldus, Microsoft, and NeXT
         Read support:
            Type    #chan   #bits   CLT?   Alpha?   Compression   Interleaving
            -----   -----   -----   ----   ------   -----------   ------------
            index   1       1       no     no       none          none
            index   1       4       no     no       none          none
            index   1       8       no     no       none          none
            index   1       8       no     no       none          none
            index   1       8       yes    no       none          none
            index   1       32      no     no       none          none
            index   1       32      yes    no       none          none
            rgb     3       8       no     no       none          none
```

```
          rgb     3      8      no     yes     none          none
          index   1      1      no     no      PackBits      none
          index   1      4      no     no      PackBits      none
          index   1      8      no     no      PackBits      none
          index   1      8      no     no      PackBits      none
          index   1      8      yes    no      PackBits      none
          index   1      32     no     no      PackBits      none
          index   1      32     yes    no      PackBits      none
          rgb     3      8      no     no      PackBits      none
          rgb     3      8      no     yes     PackBits      none
          index   1      1      no     no      LZW           none
          index   1      4      no     no      LZW           none
          index   1      8      no     no      LZW           none
          index   1      8      no     no      LZW           none
          index   1      8      yes    no      LZW           none
          index   1      32     no     no      LZW           none
          index   1      32     yes    no      LZW           none
          rgb     3      8      no     no      LZW           none
          rgb     3      8      no     yes     LZW           none
      Write support:
          Type    #chan  #bits  CLT?   Alpha?  Compression   Interleaving
          -----   -----  -----  ----   ------  -----------   ------------
          index   1      1      no     no      LZW           none
          index   1      1      no     no      PackBits      none
          index   1      1      no     no      none          none
          index   1      8      yes    no      LZW           none
          index   1      8      yes    no      PackBits      none
          index   1      8      yes    no      none          none
          index   1      8      no     no      LZW           none
          index   1      8      no     no      PackBits      none
          index   1      8      no     no      none          none
          rgb     3      8      no     yes     none          none
          rgb     3      8      no     yes     PackBits      none
          rgb     3      8      no     yes     LZW           none
          rgb     3      8      no     no      LZW           none
          rgb     3      8      no     no      PackBits      none
          rgb     3      8      no     no      none          none
```

The double **-long** output is a dump of internal tables listing what each format can and cannot handle. These tables are ordered in a way that makes sense to **imformats** and the SDSC image library, but not necessarily to human beings.

imformats also recognizes the following standard SDSC options:

-feedback
: Create a software feedback (bug report) form in the file **imformats.fbk.0**.

-fullhelp
: Display a detailed list of the arguments and how to use **imformats**.

-help
Display an abbreviated list of the arguments and how to use **imformats**.

-register
Create a software user registration form in the file **imformats.reg.0**.

-version
Display the version number and copyright for **imformats**.

All options can be abbreviated to the first few unique characters.

NOTES

Because of the length of **imformat**'s output, we recommend you pipe the output of **imformats** into a screen pager like **more**(1) or **less**(1).

The **Format** and **Description** column headings only appear if information on more than one format is displayed.

SEE ALSO

imconv(1IM), **imcopy**(1IM), **imfile**(1IM), **imflip**(1IM), **imgray**(1IM), **immono**(1IM), **impaste**(1IM), **imscale**(1IM)

For information on the SDSC image library, see **imintro**(3IM).

For information on the individual image file formats, see their respective **man** pages: **imeps**(3IM), **imgif**(3IM), **imhdf**(3IM), **imicon**(3IM), **imiff**(3IM), **immpnt**(3IM), **impbm**(3IM), **impcx**(3IM), **impgm**(3IM), **impic**(3IM), **impict**(3IM), **impix**(3IM), **impnm**(3IM), **imppm**(3IM), **imps**(3IM), **imras**(3IM), **imrgb**(3IM), **imrla**(3IM), **imrle**(3IM), **imrpbm**(3IM), **imrpgm**(3IM), **imrpnm**(3IM), **imrppm**(3IM), **imsynu**(3IM), **imtiff**(3IM), **imx**(3IM), **imxbm**(3IM), **imxwd**(3IM).

AUTHOR

Dave Nadeau
San Diego Supercomputer Center

See the individual file format **man** pages for the authors of the underlying format read and write code. The names of these **man** pages begin with the letters "im" followed by the format name. For example, the name of the TIFF **man** page is **imtiff**. To display it, enter **man imtiff**.

CONTACT

SDSC consultants, (619)534-5100, **consult@y1.sdsc.edu**

NAME

imgray - Convert an image to grayscale

SYNOPSIS

imgray [*options*] *infilename outfilename*

DESCRIPTION

imgray converts an input image to grayscale, then writes it to an output image file. If the input file contains multiple images, **imgray** converts each one and writes it to the output file. The input and output image file formats may be different.

OPTIONS

imgray has a variety of options in the following four categories:

File Selection	What input and output files to use
Format Selection	What image file format to use
Format Control	What variant of a file format to generate
Standard	Standard generic options on all SDSC tools

File Selection, *Format Selection*, *Format Control*, and *Standard* options are common to all SDSC image tools and are discussed in depth in the **man** page for **imconv**(1IM).

All options can be abbreviated to the first few unique characters.

NOTES

Conversion from color to grayscale computes the gray value for each pixel using the NTSC Y equation:

$$Gray = 0.30*R + 0.59*G + 0.11*B$$

For notes regarding file format conversion and standard image tool options, see the **man** page on **imconv**(1IM).

Error messages are reported to **stderr**.

EXAMPLES

To convert an RGB Alias **pix** file image to grayscale and store it in an **hdf** file, enter any of the following command lines:

```
imgray picture.pix picture.hdf
   or
imgray -pix picture.pix -hdf picture.hdf
   or
imgray -pix - picture.hdf < picture.pix
   or
imgray -pix picture.pix -hdf - > picture.hdf
   or
imgray -pix - -hdf - < picture.pix > picture.hdf
   or
cat picture.pix | imgray -pix - -hdf - | cat > picture.hdf
```

SEE ALSO

imconv(1IM), fBimcopy(1IM), **imfile**(1IM), **imflip**(1IM), **imformats**(1IM), **immono**(1IM), **imscale**(1IM)

For information on SDSC's image library, see **imintro**(3IM).

For information on the individual image file formats, see their respective **man** pages: **imeps**(3IM), **imgif**(3IM), **imhdf**(3IM), **imicon**(3IM), **imiff**(3IM), **immpnt**(3IM), **impbm**(3IM), **impcx**(3IM),

impgm(3IM), **impic**(3IM), **impict**(3IM), **impix**(3IM), **impnm**(3IM), **imppm**(3IM), **imps**(3IM), **imras**(3IM), **imrgb**(3IM), **imrla**(3IM), **imrle**(3IM), **imrpbm**(3IM), **imrpgm**(3IM), **imrpnm**(3IM), **imrppm**(3IM), **imsynu**(3IM), **imtiff**(3IM), **imx**(3IM), **imxbm**(3IM), **imxwd**(3IM).

AUTHOR

Dave Nadeau
San Diego Supercomputer Center

See the individual file format **man** pages for the authors of the underlying format read and write code. The names of these **man** pages begin with the letters "im" followed by the format name. For example, the name of the TIFF **man** page is **imtiff**. To display it, enter **man imtiff**.

CONTACT

SDSC consultants, (619)534-5100, **consult@y1.sdsc.edu**

NAME

immono - Convert an image to monochrome

SYNOPSIS

immono [*options*] *infilename outfilename*

DESCRIPTION

immono converts an input image to monochrome, then writes it to an output image file. If the input file contains multiple images, **immono** converts each one and writes it to the output file. The input and output image file formats may be different.

OPTIONS

immono has a variety of options in the following five categories:

File Selection	What input and output files to use
Format Selection	What image file format to use
Format Control	What variant of a file format to generate
Standard	Standard generic options on all SDSC tools
Manipulation	How the image can be manipulated

File Selection, *Format Selection*, *Format Control*, and *Standard* options are common to all SDSC image tools and are discussed in depth in the **man** page for **imconv**(1IM).

All options can be abbreviated to the first few unique characters.

Manipulation Options

Conversion of an input image to monochrome is a two-step process:

1. Convert the color image to grayscale.
2. Convert the grayscale image to monochrome.

In step 1, conversion from color to grayscale computes the gray value for each pixel using the NTSC Y equation:

$$Gray \ = \ 0.30*R + 0.59*G + 0.11*B$$

In step 2, conversion from grayscale pixel values (0-255) to monochrome pixel values (0 or 1) uses a simple thresholding technique. Pixel values equal to or higher than the threshold are considered white. Pixel values below the threshold are considered black.

By default, the threshold is set at 127. This threshold may be selected explicitly using **-threshold** followed by a positive integer.

NOTES

For notes regarding file format conversion and standard image tool options, see the **man** page on **imconv**(1IM).

Error messages are reported to **stderr**.

EXAMPLES

To convert an RGB Alias **pix** file image to monochrome and store it in a Sun **ras** file, enter any of the following command lines:

```
immono picture.pix picture.ras
   or
immono -pix picture.pix -ras picture.ras
   or
immono -pix - picture.ras < picture.pix
   or
```

```
immono -pix picture.pix -ras - > picture.ras
   or
immono -pix - -ras - < picture.pix > picture.ras
   or
cat picture.pix | immono -pix - -ras - | cat > picture.ras
```

SEE ALSO

imconv(1IM), fBimcopy(1IM), **imfile**(1IM), **imflip**(1IM), **imformats**(1IM), **imgray**(1IM), **imscale**(1IM)

For information on SDSC's image library, see **imintro**(3IM).

For information on the individual image file formats, see their respective **man** pages: **imeps**(3IM), **imgif**(3IM), **imhdf**(3IM), **imicon**(3IM), **imiff**(3IM), **immpnt**(3IM), **impbm**(3IM), **impcx**(3IM), **impgm**(3IM), **impic**(3IM), **impict**(3IM), **impix**(3IM), **impnm**(3IM), **imppm**(3IM), **imps**(3IM), **imras**(3IM), **imrgb**(3IM), **imrla**(3IM), **imrle**(3IM), **imrpbm**(3IM), **imrpgm**(3IM), **imrpnm**(3IM), **imrppm**(3IM), **imsynu**(3IM), **imtiff**(3IM), **imx**(3IM), **imxbm**(3IM), **imxwd**(3IM).

KNOWN PROBLEMS

Dithering should be supported.

AUTHOR

Dave Nadeau
San Diego Supercomputer Center

See the individual file format **man** pages for the authors of the underlying format read and write code. The names of these **man** pages begin with the letters "im" followed by the format name. For example, the name of the TIFF **man** page is **imtiff**. To display it, enter **man imtiff**.

CONTACT

SDSC consultants, (619)534-5100, **consult@y1.sdsc.edu**

NAME

impaste - Paste an image atop a background and store in a new file

SYNOPSIS

impaste [*options*] *infilename backfilename* [*outfilename*]

DESCRIPTION

impaste pastes an input image atop a background image and stores the result in a new file or back in the background image file.

If the input file contains multiple images, each input image is pasted onto a fresh copy of the background image and added to the output file. The output file contains one new image for each input image.

If the input and background files each contain multiple images, a one-for-one mapping occurs where input image #1 is pasted atop background image #1, and so on. The output file contains one new image for each input image.

The input, background, and output image file formats may be different.

OPTIONS

impaste has a variety of options in the following five categories:

File Selection	What input and output files to use
Format Selection	What image file format to use
Format Control	What variant of a file format to generate
Standard	Standard generic options on all SDSC tools
Manipulation	How the image can be manipulated

File Selection, *Format Selection*, *Format Control*, and *Standard* options are common to all SDSC image tools and are discussed in depth in the **man** page for **imconv**(1IM).

If you don't specify an output file, the background image file is treated as the output file. This allows paste-in-place operations.

The input image or the background image, but not both, may be taken from **stdin**.

All options can be abbreviated to the first few unique characters.

Manipulation Options

The pasting location within the background image is specified by:

-xposition *x*	Left edge of paste location
-yposition *y*	Top edge of paste location

The paste location may be outside the bounds of the input image. The portion of the input image that lies within the bounds of the background image will be pasted. (0,0) is the upper left corner of the image. Both positions default to values that center the incoming image left-to-right and top-to-bottom within the background image.

By giving a repeat count in the X and Y directions, the input image may be pasted repeatedly to create a tiling effect using the following:

-xrepeat *nx*	Number of times to paste horizontally
-yrepeat *ny*	Number of times to paste vertically
-xdirection *xdir*	Horizontal tiling direction

-ydirection *ydir*	Vertical tiling direction

-xrepeat and **-yrepeat** indicate how many times to paste the input image onto the same background image. A value of 1 (default) pastes the image once. Values greater than 1 repeat the paste in X and Y. A value of 0 for either repeat count requests an "infinite" repeat that tiles the image as many times as necessary to reach the horizontal or vertical edge of the background image.

-xdirection and **-ydirection** select how tiling should advance from image to image. **-xdirection** takes one of the following as its argument:

left	Tiling advances to the left
right	Tiling advances to the right

By default, tiling advances to the right.

-ydirection takes one of the following as its argument:

up	Tiling advances upward
down	Tiling advances downward

By default, tiling advances downward.

NOTES

Pasting takes place without regard to alpha planes, write protect planes, and so on. The input image is always placed pixel-for-pixel atop the background image.

Internally, an input image is converted to the same depth as the background image prior to being pasted.

Pasting of non-RGB images can cause unexpected results but is allowed. Consider this scenario: An input color index image uses color indexes 1, 2, and 3 for red, green, and blue. The background color index image uses color indexes 1, 2, and 3 as well, but its color lookup table (CLT) defines them as orange, white, and black. When the input image is pasted atop the background image, its color indexes are copied across and use the background image’s CLT. Input pixels that used to be red become orange, input green pixels become white, and input blue pixels become black because they now reference the background image’s CLT. If this is not what you want, convert the images to RGB prior to invoking **impaste**.

For notes regarding file format conversion and standard image tool options, see the **man** page on **imconv**(1IM).

Error messages are reported to **stderr**.

EXAMPLES

To paste a logo at (10,20) on a background image and save the result in a new file, use the following:

```
impaste logo.rgb background.hdf -xpos 10 -ypos 20 result.rla
```

To paste an image in the center of a background and save it back in the same background file, use the following:

```
impaste image.pix background.pix
```

To tile a pattern across the whole background, use the following:

```
impaste pattern.x -xpos 0 -ypos 0 -xrepeat 0 -yrepeat 0 background.tiff
```

To tile a pattern three times in X and none in Y, starting at (-10, -20), use the following:

```
impaste pattern.ras -xrepeat 3 -xpos -10 -ypos -20 background.ras
```

SEE ALSO

imconv(1IM), **imfile**(1IM), **imflip**(1IM), **imformats**(1IM), **imgray**(1IM), **immono**(1IM), **imscale**(1IM)

For information on SDSC's image library, see **imintro**(3IM).

For information on the individual image file formats, see their respective **man** pages: **imeps**(3IM), **imgif**(3IM), **imhdf**(3IM), **imicon**(3IM), **imiff**(3IM), **immpnt**(3IM), **impbm**(3IM), **impcx**(3IM), **impgm**(3IM), **impic**(3IM), **impict**(3IM), **impix**(3IM), **impnm**(3IM), **imppm**(3IM), **imps**(3IM), **imras**(3IM), **imrgb**(3IM), **imrla**(3IM), **imrle**(3IM), **imrpbm**(3IM), **imrpgm**(3IM), **imrpnm**(3IM), **imrppm**(3IM), **imsynu**(3IM), **imtiff**(3IM), **imx**(3IM), **imxbm**(3IM), **imxwd**(3IM).

AUTHOR

Dave Nadeau
San Diego Supercomputer Center

See the individual file format **man** pages for the authors of the underlying format read and write code. The names of these **man** pages begin with the letters "im" followed by the format name. For example, the name of the TIFF **man** page is **imtiff**. To display it, enter **man imtiff**.

CONTACT

SDSC consultants, (619)534-5100, **consult@y1.sdsc.edu**

Notes

NAME

imscale - Scale an image up or down and save it in a new file

SYNOPSIS

imscale [*options*] *infilename outfilename*

DESCRIPTION

imscale scales an input image up or down to a new size and saves the result in a new file. If the input file contains multiple images, each input image is scaled in the same way and written to the output file. The input and output image file formats may be different.

OPTIONS

imscale has a variety of options in the following five categories:

File Selection	What input and output files to use
Format Selection	What image file format to use
Format Control	What variant of a file format to generate
Standard	Standard generic options on all SDSC tools
Manipulation	How the image can be manipulated

File Selection, *Format Selection*, *Format Control*, and *Standard* options are common to all SDSC image tools and are discussed in depth in the **man** page for **imconv**(1IM).

All options can be abbreviated to the first few unique characters.

Manipulation Options

Images may be scaled in the X (horizontal) and Y (vertical) directions by specifying a scale factor or the new image size:

-xscale *xf*	Scale horizontally by factor
-yscale *yf*	Scale vertically by factor
-xsize *w*	Scale the image to a new size horizontally
-ysize *h*	Scale the image to a new size vertically

-xscale and **-yscale** each take a floating-point scale factor. To increase the size of an image, use a scale factor greater than 1.0. To decrease an image's size, use a scale factor less than 1.0. The default scale factor is 1.0 (no change).

-xsize and **-ysize** each take an integer image width or height in pixels, respectively. The incoming image is scaled up or down, as necessary, to make it the desired size. The default is to leave the image size unchanged.

These four arguments may be given alone, or in combination with the others as long as **-xscale** is not given with **-xsize**, and **-yscale** is not given with **-ysize**.

NOTES

Image scaling uses bi-linear interpolation.

Image scaling can take awhile, depending upon the size of the input image and the size of the output image.

For notes regarding file format conversion and standard image tool options, see the **man** page on **imconv**(1IM).

Error messages are reported to **stderr**.

EXAMPLES

To scale a 640x480 image up to 1280x1024 (non-uniform scaling), use any of the following:

```
imscale small.pix -xscale 2.0 -yscale 2.133 large.pix
   or
imscale small.pix -xsize 1280 -ysize 1024 large.pix
```

To reduce an image to 1/3 its original size in X, and scale it up or down to 500 pixels high in Y, use the following:

```
imscale original.rla -xscale 0.333 -ysize 500 sized.tiff
```

To copy an entire input GIF image to a PCX file, without scaling, use the following:

```
imscale incoming.gif outgoing.pcx
```

With no scaling arguments, the actions of **imscale** default to the same as the SDSC image format conversion tool **imconv**(1IM), but they take a little longer.

SEE ALSO

imconv(1IM), **imcopy**(1IM), **imfile**(1IM), **imflip**(1IM), **imformats**(1IM), **imgray**(1IM), **immono**(1IM), **impaste**(1IM)

For information on SDSC's image library, see **imintro**(3IM).

For information on the individual image file formats, see their respective **man** pages: **imeps**(3IM), **imgif**(3IM), **imhdf**(3IM), **imicon**(3IM), **imiff**(3IM), **immpnt**(3IM), **impbm**(3IM), **impcx**(3IM), **impgm**(3IM), **impic**(3IM), **impict**(3IM), **impix**(3IM), **impnm**(3IM), **imppm**(3IM), **imps**(3IM), **imras**(3IM), **imrgb**(3IM), **imrla**(3IM), **imrle**(3IM), **imrpbm**(3IM), **imrpgm**(3IM), **imrpnm**(3IM), **imrppm**(3IM), **imsynu**(3IM), **imtiff**(3IM), **imx**(3IM), **imxbm**(3IM), **imxwd**(3IM).

KNOWN PROBLEMS

Other interpolation schemes should be supported, such as pixel replication for fast power-of-two scaling (if you like jaggies).

AUTHOR

Dave Nadeau
San Diego Supercomputer Center

See the individual file format **man** pages for the authors of the underlying format read and write code. The names of these **man** pages begin with the letters "im" followed by the format name. For example, the name of the TIFF **man** page is **imtiff**. To display it, enter **man imtiff**.

CONTACT

SDSC consultants, (619)534-5100, **consult@y1.sdsc.edu**