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 six categories:

File Selection	Which input and output file to use
File Compression	How the input and output file are compressed
Format Selection	Which format to use for the input and output file
Format Control	Which variant of format to use for the output file
Miscellaneous	Other methods of altering the input or output file
Standard	Standard generic options on all SDSC tools

These categories are discussed separately below.

Any option which affects the input file begins with the prefix "-in". Any option which affects the output file begins with the prefix "-out".

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 -informat pix - newpic.ras < picture.pix
   or
imconv picture.pix -outformat ras - > newpic.ras
   or
imconv -informat pix - -outformat ras - < picture.pix > newpic.ras
   or
cat picture.pix | imconv -informat pix - -outformat 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 -informat pix and -outformat ras above. These options are discussed in more detail in the "Format Selection" section.

File Compression Options

Because image files tend to get quite large, it is often desirable to compress them with a compression program such as compress or gzip. imconv is capable of both compressing and decompressing files via the following schemes:

Suffix	External Program	Scheme Name
Z	compress	Lempel-Ziv Encoding
gz	gzip	Gnu Lempel-Ziv Encoding
z	pack	Huffman Encoding
uu	uuencode	ASCII encoding

The list above may be incomplete. For a complete list, please type:

```
imconv -help -infilecompress
```

If you would like to output a compressed file, there are two ways to do it. The first way is simply to put the suffix of the compression scheme on the end of your filename. The second way is to use the -outfilecompression flag. That is,

```
imconv infile.gif outfile.ras.Z
imconv infile.gif outfile.ras.Z -outfilecompression Z
```

will both produce a Sun Rasterfile which is Lempel-Ziv Encoded.

If the output is stdout, (i.e. the filename is '-') then the latter method must be used.

When reading a compressed file, there are three ways for imconv to know that the incoming file is compressed. imconv attempts to uncompress a file if: 1. the -infilecompression option is used, 2. the file has a magic number corresponding to the magic number for a compression scheme, or 3. The filename suffix matches one of the suffixes in the list above.

That is,

```
imconv infile.gif.uu outfile.ras
imconv infile.gif.uu -infilecompression Z outfile.ras
```

will both attempt to read a uuencoded GIF and store it as a Sun Rasterfile.

If the input is stdin, (i.e. the filename is '-'), then the -infilecompression option must be used.

The Image Tools will search for the external programs named above when doing compression or decompression, by looking in the environment PATH variable. For information about installing your own compression schemes or specifying paths for certain executables, please refer to the source code in the file imschemes.c, or the installation documentation.

Format Selection Options

imconv supports a wide variety of file formats. For a complete list of formats please use the imformats command, or refer to the man page, imintro (3IM).

For imconv to read or write an image file, it must know which format 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.

Thus, in the above example, we would read an Alias PIX file and write a Sun Rasterfile.

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 -informat pix picture.pix -outformat 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.

As mentioned above, a complete list of file formats is available via the imformats command. To obtain extensive information about each format, use imformats with the -long option, as follows:

```
imformats -long
```

You may also obtain a list of formats by using imconv's flag-specific help (described below). Simply type:

```
imconv -help -informat
```

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
```

Miscellaneous Options

Following are a list of options which affect how a file is read or how a file is written. Options which begin with "-in" affect reading of a file. Options which begin with "-out" affect writing of a file.

-inmap channel_map

The -inmap option allows you to permute the channels of an image when reading it in. A channel_map is of the form <channel>=<channel> where <channel> is one of the following:

red, green, blue, alpha, grey, index, index16, none

This option will take a channel from the image and map it to a different channel in memory. Here are some examples of how to use the -inmap option:

Given an image with red, green, blue, and alpha, create a new image such that: the alpha of the new image is the green of the original, the blue of the new image is the green of the original, the red of the new image is the alpha of the original, and there is no green in the new image:

```
imconv picture.rgb permuted.rgb -inmap alpha=green -inmap blue=green
-inmap red=alpha -inmap green=none
```

Given a greyscale picture, make an image which is red instead of grey:

```
imconv greyscale.ras red.rgb -inmap red=grey -inmap blue=none -inmap
green=none
```

Only read in the blue channel of an image, and ignore the red and green channels:

```
imconv colors.rgb -inmap red=none -inmap green=none justblue.rgb
```

To perform more specific options on components of pixels, you may use the imadjust command.

-outmap channel_map

The -outmap option allows you to permute the channels of an image when writing it out. A channel_map is of the form <channel>=<channel> where <channel> is one of the following:

red, green, blue, alpha, grey, index, index16, none

This option will take a channel from the image in memory and map it to a different channel in the output file.

For instance, to create a greyscale file whose grey values correspond to the alpha values of another file type:

```
imconv i_have_alpha.rgb greyscale.ras -outmap grey=alpha
```

To perform more specific options on components of pixels, you may use the imadjust command.

-outtransparencyrgb red green blue
-outtransparencyindex index
-outtransparency

These options affect certain types of color indexed images, such as GIF and XPM. The transparency value of an image is an index into the color lookup table which corresponds to a "transparent" color. For instance, XPM files are used as icons. By giving a transparency value, the background will show through wherever there are pixels with this value. When Mosaic or Netscape displays a GIF, the background will similarly show through wherever the transparency pixel occurs.

The -outtransparency option indicates that the most popular color in the image should be the transparency. The -outtransparencyrgb flag indicates that pixels with the given red, green, and blue values should be transparent. The -outtransparencyindex flag indicates that pixels with the given index value should be transparent.

-inpagewidth
-inpageheight
-indpi

These options are used for formats that have no predefined width and height. Since the only such format supported by the SDSC Image Library is Postscript, please refer to the man page, imps (3IM) for a description of these options.

-outgroup grouping_method

Some file formats (such as TIFF and JPEG) allow grouping of pixels by tiles instead of by scanlines. That is, normally the pixels in an image are stored in the form: row 1, row 2, row 3, etc.. However, certain compression schemes prefer that that the pixels be grouped into rectangular tiles. Then pixels are stored in the form: tile 1, tile 2, tile 3, etc...

By using the -outgroup scanlines option, you indicate the former method of storage. The -outgroup tiles option indicates the latter.

-outquality value

This option affects the quality of compression for certain compression schemes. There is often a tradeoff between the amount of compression and the quality of the resulting image. (And also the amound of time it takes to decompress or compress an image.) By specifying a high value for the quality, you indicate that you'd like the quality to be preserved. If you choose to make your image higher quality then the image will probably take up more disk space. The only format that supports the -outquality option is JPEG.

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. If this option is followed by any other options, then specific help will be given for the other options. For instance,

```
imconv -help -inmap
```

will give help about the -inmap option.

-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 -informat pix picture.pix -outformat hdf picture.hdf
   or
imconv -informat pix - picture.hdf < picture.pix
   or
imconv -informat pix picture.pix -outformat hdf - > picture.hdf
   or
imconv -informat pix - -outformat hdf - < picture.pix > picture.hdf
   or
cat picture.pix | imconv -informat pix - -outformat 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

imfile(1IM), imformats(1IM), imgray(1IM), immono(1IM)

For information on SDSC's image library, see imintro(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, Brian Duggan
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