NAME

imstoryboard - Paste images together to create a storyboard

SYNOPSIS

imstoryboard [*options*] *infilenames* -outfile *outfilename*

DESCRIPTION

imstoryboard pastes images in a storyboard, or grid, on top of a background image. The result is stored in a new file. Storyboard options to control grid margin, gutter, row, column, and other features are available. imstoryboard chooses default values to try to produce square output images.

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

OPTIONS

imstoryboard has a variety of options in the following six 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
Background Generation	Options for storyboard background selection
Storyboard Options	Parameters for storyboard formatting

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).

One 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.

Input File Names

Input file names may be given explicitly or implicitly. In an explicit list, the user gives each image file name on the command-line for imstoryboard. The following are typical invocations of imstoryboard using explicit file names:

```
imstoryboard thing1.rgb thng2.hdf sam.ras i.pix am.rla -outfile outputgroup.hdf
imstoryboard credits0.rgb credits1.rgb credits2.rgb -outfile outcredits.tiff
imstoryboard icons*.ico -outfile myicons.hdf
```

File names may be given implicitly by providing a file name template that contains the special character code "%d", like the C language printf(3) print format directive. This template will be used to generate a series of image file names by replacing the "%d" with integer frame numbers, one after another. The template myimage.%d.rgb, for instance, could generate the file names myimage.0.rgb, myimage.1.rgb, myimage.2.rgb, and so on.

The range of integer frame numbers used to generate input file names from a template is given using the -frames *range* option, where the *range* is a single value, or range of values, taking one of the following forms:

range	Generate file names with frame numbers of
n	0 through *n* (range beginning with zero)
n-m	*n* through *m* (range of values).

There is no space between *n*, the dash, and *m*. Also, *n* need not be a value less than *m*.

The following are typical invocations of imstoryboard using an implicit file name template and -frames:

```
imstoryboard credits%d.rgb -frames 0-2 -outfile outcredits.tiff
imstoryboard -frames 0-200 movie.%03d.ras -outfile collection.hdf
imstoryboard -frames 200-0 movie.%03d.ras -outfile reversed.hdf
```

Implicit and explicit file naming may be combined on the same command-line, such as:

```
imstoryboard title.rgb -frames 0-100 movie.%03d.rgb credit1.rgb copyright.rgb-outfile output.hdf
```

-frames can only be given once. If more than one implicit file name template is used on the same command-line, each use will use the same frame number range.

Image files are read in and placed into the storyboard in the order in which they appear on the command-line.

Output File Name

The output filename may be explicitly specified on the command line, such as:

```
imstoryboard title.rgb -frames 0-100 movie.%03d.rgb -outfile credit1.rgb -outfile output.hdf
```

Background Options

To use a background image located in an image file the following option is specified:

-backfile *filename*

If a background file is not specified, a background file will be generated by imstoryboard. The default background is of RGB type with a zero value for all data fields.

Background fill values are selected using one or more of the following fill options:

Option	Fill each pixel's
-red *range*	red component
-green *range*	green component
-blue *range*	blue component
-mono *range*	monochrome component
-index *range*	color index component
-hue *range*	hue component
-saturation *range*	saturation component
-intensity *range*	intensity component
-alpha *range*	alpha component

Multiple fill options may be given, as long as color spaces are not mixed. For instance, -index for filling color indexes is incompatible with -red for filling the red component of RGB triplets. Likewise, -hue for filling pixels based upon an HSI color space hue value is incompatible with -blue for filling with blue in the RGB color space. See the imintro(3IM) man page for a description of the RGB and HSI color spaces.

Each of the fill options can accept an argument giving a single value, or range of values with which to fill. Value *range* syntax takes any one of the following forms:

range	Fill with values of
n	*n* only.
n-	*n* through the maximum legal value for the component.
n-m	*n* through *m*.

There is no space between *n*, the dash, and *m*. *n* need not be a value less than *m*.

When filling with a single value, the entire background is set to the same value.

When a range of fill values are given, such as -red 0-255 or -intensity 0.6-0.2, the background will be filled with a gradation, or ramp, as selected by the -graduate option's argument:

Option	Meaning
-graduate none	Fill with a constant value
-graduate horizontal	Fill with a horizontal ramp (default)
-graduate vertical	Fill with a vertical ramp

none, horizontal, and vertical may each be abbreviated to the fewest unique letters.

The horizontal ramp for a graduated fill starts from the first fill value at the left edge of the background and goes to the second fill value at the right edge. The vertical ramp for a graduated fill goes from the first fill value at the top edge of the fill area to the second fill value at the bottom edge.

Storyboard Options

-preread may be used to storyboard images which come from files containing more than one image. With this option, each file is read twice: once to discern the number of images in the file, and once to read in the actual data.

The following options are used to control the spacing and sizing within a storyboard. Options are provided to control output image size, margin, gutter, grid frame size, and other image placement operations. If an option is not specified a default value will be assigned. The default assignments are described in the NOTES section below.

Option	Description
-xsize *x*	Width of output storyboard image, in pixels
-ysize *y*	Height of output storyboard image, in pixels
-rows *ny*	Number of grid rows for storyboard
-columns *nx*	Number of grid columns for storyboard
-xframe *x*	Width of individual grid frame, in pixels
-yframe *y*	Height of individual grid frame, in pixels
-scale *scaleValue*	Fractional amount to scale both grid frame axes
-xscale *xScaleValue*	Fractional amount to scale grid frame width
-yscale *yScaleValue*	Fractional amount to scale grid frame height
-xmargin *x*	Width for left and right margins, in pixels
-ymargin *y*	Height for top and bottom margins, in pixels
-xgutter *x*	Width to place between storyboard images, in pixels
-ygutter *y*	Height to place between storyboard images, in pixels

Each of the input images will be scaled to the frame size given by -xframe and -yframe.

The scale options conflict with the -xframe and -yframe, and therefore cannot be used together.

NOTES

If an option is not given on the command-line, one of the following defaults will be assigned as the option's value:

Rows and Columns

If -rows and -columns are not specified, then the most square array of rows and columns, based on the number of input files, will be computed. These formula will determine the row and column defaults:

rows = floor(sqrt(# of input files+0.5))
columns = ceil(sqrt(# of input files))

Otherwise when -rows or -columns, but not both, are specified the defaults will assign a value large enough to hold all of the input images.

The row and column defaults assume single image input files. The number of input files is used to calculate the row and column defaults for the storyboard. This action may lead to incorrect row and column defaults when an input file contains more than one image.

The following outline pertains to both x and y axis options:

If output size not given
    frame = size of first image
    margin = 10% of frame size
    gutter = 10% of frame size
    output size = enough space to hold above
Else output size given
    margin = 10% of output size / (rows or columns)
    gutter = 10% of output size / (rows or columns)
    frame = what ever is left over at this point

scale = 1.0

Images placed into the storyboard will be scaled to the grid frame size. The user should select a frame size, or scale factor in order to preserve the image's aspect ratio.

imstoryboard uses the C language sprintf(3) routine to replace the file name template's "%d" with successive image frame numbers. The use of sprintf(3) means the full range of "%d" format directive syntax is available in generating input file names. While the sprintf(3) man page has full details on the "%d" print directive, the following is a summary as it applies to input file naming:

The "%d" directive has the following general syntax:

"%" [*flags*] [*digit*] ["." *precision*] "d"

The directive always starts with a "%" character and ends with a "d" character. Field names between [ ]s in the syntax above are optional. With no optional field value, the simplest form of the directive is just "%d".

flags Following the "%" is an optional set of *flags* characters:

flag	Meaning
-	Left justify the frame number. Default is right justified.
+	Include the sign (+ or -) of the frame number. Default is just -.
(blank)	Use a blank instead of a + for positive frame numbers.

Except for peculiar file naming, users of imstoryboard can omit the optional *flags* portion of a print directive.

digit Following the *flags* field is an optional *digit* field that gives the field width of the number, in characters. If the number requires fewer than this minimum, it will be padded on the right or left with blanks (padding is controlled by the - *flag* above). If the number requires more than this minimum, it will be taken.

The *digit* field can be preceded by a 0 (zero) to cause padding to be done with leading zeroes instead of blanks. This is useful when referring to frame numbers of the form "001," "002," "003," and so on.

precision

The *digit* field may be optionally followed by a "." (period) and a *precision* number that gives the minimum number of digits for the frame number. This isn't particularly useful for users of imstoryboard and can be skipped.

The following are a few typical "%d" directives for use in imstoryboard:

Directive	Use in imstoryboard
%d	No leading zeros or blanks. Generates numbers like "0," "1," "2," "3," etc.
%03d	Leading zeros to fill 3 characters, including the number. Generates numbers like "000," "001," "002," "003," etc.
%.3d	Same effect as "%03d"

Storyboarding 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 in the storyboard.

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

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 imstoryboard.

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

Storyboard four images on a background and save the storyboard in a new HDF file:

```
imstoryboard sdsc.pix boris.rgb brs.rgb im52.rgb -outfile result.hdf
```

Storyboard the same four images scaled 50%:

```
imstoryboard sdsc.pix boris.rgb brs.rgb im52.rgb -outfile result.hdf -scale 0.5
```

Storyboard the same four images on a graduated red background:

```
imstoryboard sdsc.pix boris.rgb brs.rgb im52.rgb -graduate vertical -red 255-0 -outfile result.hdf
```

Storyboard the same four images on a background RGB file:

```
imstoryboard sdsc.pix boris.rgb brs.rgb im52.rgb -backfile background.rgb -outfile result.hdf
```

Storyboard two images with a left and right margin of 30 pixels

```
imstoryboard house.gif garage.gif -xmargin 30 -outfile result.rgb
```

Storyboard two images with a grid frame height of 100 pixels and grid frame width of 200 pixels

imstoryboard stereoL.rgb stereoR.rgb -xframe 200 -yframe 100 -outfile result.rle

Storyboard four images in one column:

imstoryboard sdsc.pix boris.rgb brs.rgb im52.rgb -columns 1 -outfile result.hdf

SEE ALSO

imcat (1IM), imfill (1IM), impaste (1IM), imscale (1IM), ImVfbCopy (3IM) ImVfbFill (3IM) ImVfbResize (3IM)

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

AUTHOR

Jason Secosky, 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