

NAME

sips — scriptable image processing system.

SYNOPSIS

sips [*image-functions*] *imagefile* . . .
sips [*profile-functions*] *profile* . . .

DESCRIPTION

This tool is used to query or modify raster image files and ColorSync ICC profiles. Its functionality can also be used through the "Image Events" AppleScript suite. It also supports executing JavaScript to either modify or generate images.

FUNCTIONS

Profile query functions:

-g *key*
--getProperty *key*
Output the property value for key to stdout.

-x *tag tagFile*
--extractTag *tag tagFile*
Write a profile tag element to tagFile.

--verify
Verify any profile problems and log output to stdout.

-l
--oneLine
Output for each file on a single line with tab delimiter

Image query functions:

-g *key*
--getProperty *key*
Output the property value for key to stdout.

-x *profile*
--extractProfile *profile*
Get the embedded profile from image and write it to profile.

-l
--oneLine
Output for each file on a single line with tab delimiter

Profile modification functions:

-s *key value*
--setProperty *key value*
Set a property value for key to value.

-d *key*
--deleteProperty *key*
Remove a property value for key.

--deleteTag *tag*
Remove the tag element from a profile.

--copyTag *srcTag dstTag*
Copy the srcTag element of a profile to dstTag.

--loadTag *tag tagFile*
Set the tag element of a profile to the contents of tagFile.

--repair
Repair any profile problems and log output to stdout.

-o *file-or-directory*
--out *file-or-directory*

Image modification functions:

-s *key value*
--setProperty *key value*
Set a property value for key to value.

-d *key*
--deleteProperty *key*
Remove a property value for key.

-e *profile*
--embedProfile *profile*
Embed profile in image.

-E *profile*
--embedProfileIfNone *profile*
Embed profile in image only if image doesn't have a profile.

-m *profile*
--matchTo *profile*
Color match image to profile.

-M *profile intent*
--matchToWithIntent *profile intent*
Color match image to profile with rendering intent perceptual | relative | saturation | absolute.

--deleteColorManagementProperties
Delete color management properties in TIFF, PNG, and EXIF dictionaries.

-r *degreesCW*
--rotate *degreesCW*

-f *horizontal/vertical*
--flip *horizontal/vertical*

-c *pixelsH pixelsW*
--cropToHeightWidth *pixelsH pixelsW*
Crop image to fit specified size.

--cropOffset *offsetY offsetH*
Crop offset from top left corner.

-p *pixelsH pixelsW*
--padToHeightWidth *pixelsH pixelsW*
Pad image with pixels to fit specified size.

--padColor *hexcolor*
Use this color when padding. White=FFFFFF, Red=FF0000, Default=Black=000000

-z *pixelsH pixelsW*
--resampleHeightWidth *pixelsH pixelsW*
Resample image at specified size. Image aspect ratio may be altered.

--resampleWidth *pixelsW*
Resample image to specified width.

--resampleHeight *pixelsH*
Resample image to specified height.

-Z *pixelsWH*
--resampleHeightWidthMax *pixelsWH*
Resample image so height and width aren't greater than specified size.

-i
--addIcon
Add a Finder icon to image file.

--optimizeColorForSharing
Optimize color for sharing.

-o *file-or-directory*
--out *file-or-directory*

-j *file*
--js *file*
Execute JavaScript file

Other functions:

--debug
Enable debugging output

-h
--help
Show help

-H
--helpProperties
Show help for properties

--man
Generate man pages

-v
--version
Show the version

--formats
Show the read/write formats

JavaScript

HTML Canvas objects can be created and used to create a 2D drawing context. The commands for drawing into the context are well documented elsewhere. This section will describe the sips global object and other interesting classes.

Global variable (sips) properties

images

Valid images passed as arguments converted into an array of Image objects

arguments

Arguments passed into the program as an array of strings

size Recommended size for output. Setting the crop or resample flags will set this value.

longestEdge

If specified, the value of the `-Z/--resampleHeightWidthMax` option. [default: 0]

outputPath

Output directory [default: current directory]

Image Object

name Name of image

size Size of image (pixels)

properties

Image properties

getProperty(name)

Return the image property for name, if any.

sizeToFitLongestEdge(length)

Return the size that will contain the image with the longest edge set to length. Maintains aspect ratio.

Output Object

new Output(context, name[, type])

Output the context to disk with name and optional type (extension or UTI).

addToQueue()

Adds the output to the queue to be written to disk.

Functions

print(str)

Output to standard output. Equivalent to `console.log(str)`.

OPTIONS

The commands `--getProperty`, `--setProperty`, and `--deleteProperty` can use one of the following keys as a parameter.

Special property keys:

all binary data

allxml binary data

Image property keys:

dpiHeight float

dpiWidth float

pixelHeight integer (read-only)

pixelWidth integer (read-only)

typeIdentifier string (read-only)

format string jpeg | tiff | png | gif | jp2 | pict | bmp | qtif | psd | sgi | tga

formatOptions string default | [low|normal|high|best|<percent>] | [lzw|packbits]

space string (read-only)

samplesPerPixel integer (read-only)

bitsPerSample integer (read-only)

creation string (read-only)

make string

model string

software string (read-only)

description string

copyright string

artist string

profile	binary data
hasAlpha	boolean (read-only)
Profile property keys:	
description	utf8 string
size	integer (read-only)
cmm	string
version	string
class	string (read-only)
space	string (read-only)
pcs	string (read-only)
creation	string
platform	string
quality	string normal draft best
deviceManufacturer	string
deviceModel	integer
deviceAttributes0	integer
deviceAttributes1	integer
renderingIntent	string perceptual relative saturation absolute
creator	string
copyright	string
md5	string (read-only)