# ImageRotate

**ImageRotate** [***flags***] ***imageMatrix***

The ImageRotate operation rotates the image clockwise by *angle* (degrees) or counter-clockwise if /W is used.

The resulting image is saved in the wave M_RotatedImage unless the /O flag is specified. The size of the resulting image depends on the angle of rotation.

The portions of the image corresponding to points outside the domain of the original image are set to the default value 64 or the value specified by the /E flag.

You can apply ImageRotate to 2D and 3D waves of any data type.

**Flags**

/A=*angle* Specifies the rotation angle measured in degrees in the counter-clockwise direction. For rotations by exactly 90 degrees use /C or /W instead.

/C Specifies a counter-clockwise rotation by 90 degrees.

/E= *val* Specifies the value for pixels that are outside the domain of the original image. By default pixels are set to 64. If you specify `/E=(NaN)` and your data is of type char, short, or long, the operation sets the external values to 64.

/F Rotates image by 180 degrees.

/H Flip the image horizontally.

/O Overwrites the original image with the rotated image.

/Q Quiet mode. Without this flag the operation writes warnings in the history area.

/RGBA=[*R*, *G*, *B* [, *A*])

Specifies the RGB or RGBA values of pixels that lie outside the domain occupied by the original image. This flag was added in Igor Pro 7.00.

/S Uses source image wave scaling to preserve scaling and relative locations of objects in the image for rotation angles that are multiples of 90 degrees.

/V Flip the image vertically.

/W Specifies a clockwise rotation by 90 degrees.

/Z Ignore errors.

**See Also**

The **MatrixTranspose** operation.

# ImageSave

**ImageSave** [***flags***] ***waveName*** [[**as**]***fileNameStr***]

The ImageSave operation saves the named wave as an image file.

Previously this operation used QuickTime to implement the saving of some file formats. As of Igor Pro 7.00, it no longer uses QuickTime. Consequently, some file formats re no longer supported and some flags have changed.

**Parameters**

The file to be written is specified by *fileNameStr* and /P=*pathName* where pathName is the name of an Igor symbolic path. *fileNameStr* can be a full path to the file, in which case /P is not needed, a partial path relative to the folder associated with *pathName*, or the name of a file in the folder associated with *pathName*. If Igor can not determine the location of the file from *fileNameStr* and *pathName*, it displays a dialog allowing you to specify the file.

If you use a full or partial path for *fileNameStr*, see **Path Separators** on page III-451 for details on forming the path.

If you want to force a dialog to select the file, omit the *fileNameStr* parameter or specify "" for *fileNameStr* or use the /I flag. The "as" keyword before *fileNameStr* is optional.

In Igor Pro 7.00 or later, if you specify "Clipboard" for *fileNameStr* and the file type is JPEG or PNG, Igor writes the image to the clipboard.

In Igor Pro 7.00 or later, if the file type is PNG or JPEG you can write the image to the picture gallery using the magic path name _PictGallery_. For example

```
ImageSave/T=PNG/P=_PictGallery_ waveName
```

**Flags**

/ALPH=*mode* Sets the value used for the ExtraSamples tag (338) in the TIFF file format. This flag was added in Igor Pro 7.00.

The tag tells a TIFF file reader how to use the alpha values. Supported modes are 1 for premultiplied and 2 for unassociated alpha. Premultiplied means that the RGB components have already been multiplied by alpha. Unassociated means that the RGB components are raw.

By default, alpha is assumed to be premultiplied (*mode*=1).

/BIGT Saves TIFF files in BigTIFF format, a newer TIFF standard that allows you to save large images (over 2GB) and provides support for several compression formats. Many applications that load TIFF files do not support BigTIFF.

The /BIGT flag was added in Igor Pro 9.00.

/C=*comp* Specifies the compression to use when saving BigTIFF:

| *comp* | **BigTIFF Compression** |
|---|---|
| 1 | No compression (default) |
| 2 | CITTRLE |
| 3 | CCITTFAX3 |
| 4 | CCITTFAX4 |
| 5 | LZW |
| 7 | JPG |
| 8 | ADOBE_DEFLATE |
| 32773 | PACKBITS |
| 32946 | DEFLATE |
| 32947 | DCS |
| 34712 | JP2000s |

The /C flag was added in Igor Pro 9.00.

/D=*depth* This flag is deprecated and should not be used in new code.

Specifies color depth in bits-per-pixel. Integer values of 1, 8, 16, 24, 32, and 40 are supported. A *depth* of 40 specifies 8-bit grayscale; a *depth* of 8 saves the file as 8-bits/pixel with a color lookup table. If /D is omitted, it acts like /D=8.

Saving with a color table may cause some loss of fidelity.

See also the discussion of saving TIFF files below.

/DS=*depth* Saves TIFF data in *depth* bits/sample.

Values for *depth* of 8, 16, 32 and 64 are supported. The default is 8.

The total number of bits/pixel is: (bits/sample) * (samples/pixel).

When using 32 or 64 bits/sample, *srcWave* is saved without normalization.

/F — This flag is deprecated and should not be used in new code.

Saves the wave as single precision float. The data is not normalized. Applies only to TIFF files.

/I — Interactive mode. Forces ImageSave to display a Save File dialog, even if the file and folder location are fully specified.

/IGOR — In Igor6 this flag told Igor to use internal code rather than QuickTime to write TIFF files. Igor no longer uses QuickTime and always uses internal code. This flag is still accepted but has no effect.

/O — Overwrites the specified file if it exists. If /O is omitted and the file exists, ImageSave displays a Save File dialog.

/P=*pathName* — Specifies the folder in which to save the image file. *pathName* is the name of an existing symbolic path.

/Q=*quality* — Specifies the quality of the compressed image. *quality* is a number between 0 and 1, with 1 being the highest quality. This is only applicable when saving in formats with lossy compression like JPEG.

/S — Saves as stack. Applies only to 3D or 4D source waves saved as TIFF files.

/T=*fileTypeStr* — Specifies the type of file to be saved.

| *fileTypeStr* | Saved Image Format |
|---|---|
| "jpeg" | JPEG file |
| "png" | PNG file |
| "rpng" | raw PNG file (see **Saving as Raw PNG**) |
| "tiff" | TIFF file |

If /T is omitted, the default type is "tiff".

/U — Prevents normalization. This works when saving TIFF only. See **Saving as TIFF** below.

/WT=*tagWave* — Saves TIFF files with file tags as specified by *tagWave*, which is a text wave consisting of a row and 5 columns for each tag (see description of the /RAT flag under **ImageLoad**). It ignores any information in the second column but will write the tags sequentially (only in the first Image File Directory (IFD) if there is more than one image). If the fourth column contains a negative number, there must be a wave, whose name is given in the fifth column of *tagWave*, in the same data folder as *tagWave*. You must make sure that: (1) tag numbers are legal and do not conflict with any existing tags; (2) the data type and data size are consistent with the amount of data saved in external waves.

/Z — No error reporting.

**Details**

Image files are characterized by the number of color components per pixel and the bit-depth of each components. Igor displays images that consist of 1 components (gray-scale/false color), 3 components (RGB) or 4 components (RGBA) per pixel. You can display waves of all real numeric types as images, but wave data are assumed to be either in the range of [0,255] for 8-bits/components or [0,65535] for all other numeric types. For more information see **Creating an Image Plot** on page II-386.

When you save a numeric wave as image file your options depend on the number of components (layers) of your wave and its number type.

**Saving as PNG**

You can save an Igor wave as a PNG file with 24 bits per pixel or 32 bits per pixel. The following table describes how the wave's data type corresponds to the PNG pixel format.

| Source Wave | PNG Pixel Format |
|---|---|
| 1-layer any type | RGB 24 bits per pixel gray normalized [0,255] |
| 3-layer unsigned byte | RGB 24 bits per pixel no scaling |
| 3-layer unsigned short | RGB 24 bits per pixel data divided by 256 |
| 3-layer all other types | RGB 24 bits per pixel data normalized [0,255] |
| 4-layer unsigned byte | RGBA 32 bits per pixel no scaling |
| 4-layer unsigned short | RGBA 32 bits per pixel data divided by 256 |
| 4-layer all other types | RGBA 32 bits per pixel data normalized [0,255] |

Gray means that the three components of each pixel have the identical value. Data normalization consists of finding the global (over all layers) minimum and maximum values of *srcWave* followed by scaling to the full range, e.g.,

```
minValue = WaveMin(srcWave)
maxValue = WaveMax(srcWave)
outValue[i][j][k] = 255*(srcWave[i][j][k]-minValue)/(maxValue-minValue)
```

**Saving as Raw PNG**

The rpng image format requires a wave in 8- or 16-bit unsigned integer format with 1 to 4 layers. Use one layer for grayscale, 3 layers for rgb color, and the extra layer for an alpha channel. If X or Y scaling is not unity, they both must be valid and must be either inches per pixel or meters per pixel. If the units are not inches they are taken to be meters.

**Saving as TIFF**

ImageSave supports saving uncompressed TIFF images of 8, 16, 32 and 64 bits per color component with 1, 3 or 4 components per pixel.

Depending on the data type of your wave and the depth of the image file being created, ImageSave may save a "normalized" version of your data. The normalized version is scaled to fit in the range of the image file data type. For example, if you save a 16-bit Igor wave containing pixel values from -1000 to +1000 in an 8-bit grayscale TIFF file, ImageSave will map the wave values -1000 and +1000 to the file values 0 and 255 respectively. When saving an image file of 16-bits/component, Igor normalizes to 65535 as the full-scale value.

There is no normalization when you save in floating point (32 or 64 bits/component). Normalization is also not done when saving 8-bit wave data to an 8-bit image file. You can disable normalization with the /U flag.

Saving in floating point can lead to large image files (e.g., 64-bit/component RGBA has 256 bits/pixel) which are not supported by many applications.

**Saving 3D or 4D Waves as a Stack of TIFF Images**

If your Igor data is a 3D wave other than an RGB wave or a 4D wave, you can save it as a stack of grayscale images without a color map.

Use /S to indicate that you want to save a stack of images rather than a single image from the first layer of the wave.

Use /D=8 to save as 8 bits. Normalization is done except if the wave data is 8 bits.

Use /D=16 to save as 16 bits with normalization.

Use /F to save as single-precision floating point without normalization. Many programs can not read this format.

Use /U to prevent normalization.

Stacked images are normalized on a layer by layer basis. If you want to have uniform scaling and normalization you should convert your wave to the file data type before executing ImageSave.

**See Also**

The **ImageLoad** operation for loading image files into waves.