Parameter Specification	Object Colored
`alblRGB(`*axisName*`)=(`*r*`,`*g*`,`*b*`[,`*a*`])`	Axis labels
`axRGB(`*axisName*`)=(`*r*`,`*g*`,`*b*`[,`*a*`])`	Axis
`cbRGB=(`*r*`,`*g*`,`*b*`[,`*a*`])`	Control bar background
`gbRGB=(`*r*`,`*g*`,`*b*`[,`*a*`])`	Graph background
`gbGradient`	See **Gradient Fills** on page III-498 for details.
`gbGradientExtra`	See **Gradient Fills** on page III-498 for details.
`gridRGB(`*axisName*`)=(`*r*`,`*g*`,`*b*`[,`*a*`])`	Axis grid lines
`tickRGB(axisName )=(`*r*`,`*g*`,`*b*`[,`*a*`])`	Axis Tick marks
`tlblRGB(`*axisName*`)=(`*r*`,`*g*`,`*b*`[,`*a*`])`	Axis Tick labels
`wbRGB=(`*r*`,`*g*`,`*b*`[,`*a*`])`	Window background
`wbGradient`	See **Gradient Fills** on page III-498 for details.
`wbGradientExtra`	See **Gradient Fills** on page III-498 for details.

Flags

/W=*winName*	Modifies the named graph window or subwindow. When omitted, action will affect the active window or subwindow. This must be the first flag specified when used in a Proc or Macro or on the command line. When identifying a subwindow with *winName*, see **Subwindow Syntax** on page III-92 for details on forming the window hierarchy.
/Z	Does not generate an error if the named axis does not exist in a style macro.

Details

On Windows, use maximum white to set the control bar background color to track the 3D Objects color in the Appearance Tab of the Display Properties control panel:

```
ModifyGraph cbRGB=(65535,65535,65535)
```

See Also

See **Instance Notation** on page IV-20.

ModifyImage

`ModifyImage` `[`***`/W=winName`***`]` ***`imageInstance, keyword = value`***
`[`***`, keyword = value`***`]…`

The ModifyImage operation changes properties of the given image in the top graph (or the specified graph if /W is used). *imageInstance* is the name of the image to be altered. This name is usually simply the name of the matrix wave containing the image data. If the same matrix wave is displayed more than once, you must append #0, #1 etc. to the name to distinguish which is which.

imageInstance can also take the form of a null name with an instance number to affect the instanceth image. That is,

```
ModifyImage ''#1
```

modifies the appearance of the second image that was appended to the top graph, no matter what the image names are. Note: two single quotes are used, not a double quote.

Parameters

Here are the keyword-value pairs. These apply to false color images in which the data in the matrix is used as an index into a color table. They do not apply to direct color images in which the data in the matrix specifies the color directly.

cindex=*matrixWave* Sets the Z value mapping mode such that image colors are determined by doing a lookup in the specified matrix wave.

matrixWave is a 3 column wave that contains red, green, and blue values from 0 to 65535. (The matrix can actually have more than three columns. It ignores any extra columns.)

The color at Z=z is determined by finding the RGB values in the row of *matrixWave* whose scaled X index is z. In other words, the red value is *matrixWave*(z)[0], the green value is *matrixWave*(z)[1] and the blue value is *matrixWave*(z)[2].

If *matrixWave* has default X scaling, where the scaled X index equals the point number, then row 0 contains the color for Z=0, row 1 contains the color for Z=1, etc.

If you use cindex, you should not use ctab in the same command.

ctab={*zMin*, *zMax*, *ctName*, *reverse*}

Sets the z mapping mode by which values in the matrix are mapped linearly into the color table specified by *ctName*.

zMin and *zMax* set the range of z values to map. Omit *zMin* or *zMax* to leave as is or use * to autoscale.

The color table name can be missing if you want to leave it as is.

ctName can be any color table name returned by the CTabList function, such as Grays or Rainbow (see **Image Color Tables** on page II-392) or the name of a 3 column or 4 column color table wave (**Color Table Waves** on page II-399).

A color table wave name supplied to `ctab` must not be the name of a built-in color table (see **CTabList**). A 3 column or 4 column color table wave must have values that range between 0 and 65535. Column 0 is red, 1 is green, and 2 is blue. In column 3 a value of 65535 is opaque, and 0 is fully transparent.

Set *reverse* to 1 to reverse the color table. Setting it to 0 or omitting it leaves the color table unreversed.

ctabAutoscale=*autoBits*

Sets the range of data used for autoscaling ctab * values.

Bit 0:	Autoscales only the XY subset being displayed.
Bit 1:	Autoscales only the current plane being displayed.

If neither bit is set (if *autoBits* = 0, the default), then all of the data in the image wave is used to autoscale the *'d *zMin*, *zMax* values for ctab.

eval={*value*, *red*, *green*, *blue*, [*alpha*]}

If the red, green, and blue values are in the valid range for a color value (0 to 65535) the explicit value-color pair is added (or updated if value already exists). If the color values are out of range (-1 is suggested) then the value is removed from the list if it is present (no error if it is not).

alpha is optional: a value of 65535 is opaque, and 0 is fully transparent.

explicit=1 *or* 0	Turns explicit (monochrome) mode on (1) or off (0). Meant to be used with unsigned byte data but will do the best it can for other types. If value of data is equal to one of the defined explicit values then its defined color is used otherwise the pixel will be blank. The default predefined values are: 255: black 0: white You can add, change, or delete explicit values with the eval keyword.
genericNotRGB=*v*	Controls the auto-detection of three and four plane images: *v*=1: Suppresses the auto-detection of three and four plane images as direct color (RGB or RGBA), like AppendImage/G=1. *v*=0: Three and four plane images are treated as direct color (RGB or RGBA). This is the default if genericNotRGB is omitted. The genericNotRGB keyword was added in Igor Pro 9.00.
imCmplxMode=*m*	Sets complex data display mode. *m*=0: Magnitude (default). *m*=1: Real only. *m*=2: Imaginary only. *m*=3: Phase in radians.
interpolate= *mode*	*mode* = 1 turns on smoothing of the boundaries between pixels. Since this is implemented via system graphics calls and not by Igor actually doing the interpolation, it will not affect EPS or EMF export on Windows and will not affect EPS export on Mac. Although this may create a more esthetically pleasing display, it is not clear that it is appropriate for scientific data. *mode* = -1 forces pixels to be drawn as individual rectangles. This is sometimes needed when a third-party program improperly interpolates PDF or EPS exported images.
log= 1 or 0	0 sets the default linearly-spaced false-image colors. 1 turns on logarithmically-spaced false-image colors. This requires that the image values be greater than 0 to display correctly. Affects the image colors for color table and color index images only (see **Color Table Details** on page II-395 and **Indexed Color Details** on page II-400).
lookup= *waveName*	Specifies an optional 1D wave that can be used to modify the mapping of scaled z values into the color table specified with the ctab parameter. Values should range from 0.0 to 1.0. A linear ramp from 0 to 1 would have no effect while a ramp from 1 to 0 would reverse the image. Used to apply gamma correction to grayscale images or for special effects. Use a NULL wave (`$""`) to remove the option.
maxRGB=(*red*, *green*, *blue*, [*alpha*])	
	Sets the color of image values greater than the ctab *zMax* or greater than the cindex of the *matrixWave* maximum X scaling value. Also turns max color mode on. The *red*, *green*, and *blue* color values are in the range of 0 to 65535. *alpha* is optional: a value of 65535 is opaque, and 0 is fully transparent.
maxRGB=1 *or* 0 *or* NaN	

Turns max color mode off, on, or transparent. These modes affect the display of image values greater than the ctab *zMax* or greater than the cindex of the *matrixWave* maximum X scaling value.

1:	Turns on max color mode. The color of the affected image pixels is black or the last color set by maxRGB=(*red*, *green*, *blue*).
0:	Turns off max color mode (default). The color of the affected image pixels is the last color table or color index color.
NaN:	Transparent max color mode. The affected image pixels are not drawn.

minRGB=(*red*, *green*, *blue*, [*alpha*])

Sets the color of image values less than the ctab *zMin* or less than the cindex of the *matrixWave* minimum X scaling value. Also turns min color mode on.

The *red*, *green,* and *blue* color values are in the range of 0 to 65535.

alpha is optional: a value of 65535 is opaque, and 0 is fully transparent.

minRGB=1 *or* 0 *or* NaN

Turns min color mode off, on, or transparent. These modes affect the display of image values less than the ctab *zMin* or less than the cindex of the *matrixWave* minimum X scaling value.

1:	Turns on min color mode. The color of the affected image pixels is black or the last color set by minRGB=(*red*, *green*, *blue*).
0:	Turns off min color mode (default). The color of the affected image pixels is the first color table or color index color.
NaN:	Transparent min color mode. The affected image pixels are not drawn.

plane=*p* Determines which part of a 3D or 4D image wave to display.

The meaning of *p* depends on the nature of the image wave. If the size of the layer dimension of the image wave is exactly three then the wave is treated as RGB data with R, G, and B data in the three layers. If the size of the layer dimension is exactly four, then the wave is treated as RGBA data, with A in the fourth layer. Otherwise each layer of the wave is treated as a separate grayscale image.

Plane=p With RGB Data

If the wave is 3D, plane=*p* has no effect.

If the wave is 4D, each chunk contains a different set of R, G and B layers and *p* selects which chunk to display.

Plane=p With Grayscale Data

If the wave is 3D, *p* selects which layer to display.

If the wave is 4D, plane=*p* acts as if all of the chunks were combined into a virtual 3D wave and *p* selects which layer of this virtual 3D wave to display.

rgbMult=*m* If *m* is non-zero, direct color values in a 3-plane RGB or 4-plane RGBA image are multiplied by *m*. (Alpha values are not multiplied.) This would typically be used for 10, 12 or 14 bit integers in a 16 bit word. For example, if your image data is 14 bits, use rgbMult=4.