.ColorMaskApply (PHOTO-PAINT)

This command applies the selected color mask. Unlike a normal mask, the mask marquee is not visible and the color mask is not editable.

Syntax

.ColorMaskApply .MaskMode=*long*, .Smoothing=*long*, .ToleranceMode=*long*

Syntax	Definition
.MaskMode	Specifies the mode of the Color Mask. 0 = Protect 1 = Modify
.Smoothing	Specifies the amount of smoothing to be applied by the Color Mask. Valid values range from 0 to 100.
.ToleranceMode	Color Tolerance is an adjustment for controlling how many shades of a selected color are masked. Selecting a higher tolerance value causes a wider range of colors to be masked; a lower tolerance value causes a smaller range of colors to be masked. 0 = Normal 1 = HSB 2 = Hue 3 = Saturation 4 = Brightness

Example

```
.ColorMaskApply 1, 0, 0
```

The above example applies a modified normal color mask.

{button ,AL(`PP_ColorMask_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.ColorMaskColor (PHOTO-PAINT)

This command sets the color mode used for the color mask.

Syntax

.ColorMaskColor .Number=*long*, .ColorModel=*long*, .Color1=*long*, .Color2=*long*, .Color3=*long*, .Color 4=*long*, .Normal=*long*, .Hue=*long*, .Saturation=*long*, .Brightness=*long*

Syntax	Definition
.Number	Identifies the colors in the Color Mask. Valid values range from 1 to 10. Refer to the Color Mask roll-up for more details.
.ColorModel	Specifies the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Specifies the first color component for .ColorModel. For example, Red is the first color component for RGB. Click + for valid value ranges.
.Color2	Specifies the second color component for .ColorModel. For example, Green is the second color component for RGB. Click + for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.

.Color3 Specifies the third color component for .ColorModel. For example, Blue is the third color component for RGB. Click + for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.

.Color4	Specifies the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. Click + for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.

.Normal Set to 0 to enable normal mode. Set to 1 to enable HSB mode.
Note: If Normal mode is selected, set Hue, Saturation, and Brightness to 0.

.Hue Specifies the hue. In the HSB color model, hue is the main attribute in a color that distinguishes it from other colors. Blue, green and red, for example, are all hues.

.Saturation	Specifies the saturation. Saturation is the purity of a color. The HSB color model uses Saturation as a component that determines the purity or intensity of a color. The more colors used to mix a color, the duller the color looks.
.Brightness	Specifies the brightness. In the HSB color model, the component that determines the amount of black in a color.

Example

```
.ColorMaskColor 3, 5, 0, 0, 255, 0, 10, 10, 10, 10
```

The above example sets the color mask color to blue.

{button ,AL(`PP_ColorMask_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.ColorMaskCreateMask (PHOTO-PAINT)

This command creates a mask from a color mask.

Syntax

.ColorMaskCreateMask .DrawMode=*long*, .MaskMode=*long*, .Smoothing=*long*, .ToleranceMode=*long*

Syntax	Definition
.DrawMode	Specifies the draw mode. 0 = Normal 1 = Add 2 = Subtract 3 = XOR
.MaskMode	Specifies the mode of the Color Mask. 0 = Protect 1 = Modify
.Smoothing	Specifies the amount of smoothing to be applied by the Color Mask. Valid values range from 0 to 100.
.ToleranceMode	Color Tolerance is an adjustment for controlling how many shades of a selected color are masked. Selecting a higher tolerance value causes a wider range of colors to be masked; a lower tolerance value causes a smaller range of colors to be masked. 0 = Normal 1 = HSB 2 = Hue 3 = Saturation 4 = Brightness

Note

The .ColorMaskCreateMask command must be preceded by the .ColorMaskApply command.

Example

```
.ColorMaskCreateMask 0, 1, 0, 0
```

The above example creates a color mask.

{button ,AL(`PP_ColorMask_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.ColorMaskLoad (PHOTO-PAINT)

This command loads the colors and tolerance settings for the selected color mask from a color mask file.

Syntax
.ColorMaskLoad .FileName=*string*

Syntax	Definition
.FileName	Specifies the file name of the Color Mask to be loaded. Color Mask files have .CMK extensions.

Note

You will need to use .ColorMaskApply to set the other Color Mask attributes.

Example

```
.ColorMaskLoad "MASK1.CMK"
```

The above example loads the color mask named "MASK1.CMK".

{button ,AL(`PP_ColorMask_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.ColorMaskRemove (PHOTO-PAINT)

This command removes the current color mask.

Syntax
.ColorMaskRemove

Note

The .ColorMaskRemove command must be preceded by the .ColorMaskApply and .ColorMaskCreateMask commands.

Example

```
.ColorMaskRemove
```

The above example removes the active color mask.

{button ,AL(`PP_ColorMask_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.ColorMaskReset (PHOTO-PAINT)

This command resets the color mask to its original default settings. Tolerance values are set to 10, colors are set to white, and all colors are inactivated.

Syntax
.ColorMaskReset

Example
```
.ColorMaskReset
```

The above example resets the color mask to its original default settings.

{button ,AL(`PP_ColorMask_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.ColorMaskSave (PHOTO-PAINT)

This command saves the current color mask settings in a color mask file.

Syntax
.ColorMaskSave .FileName=*string*

Syntax	Definition
.FileName	Specifies the file name of the Color Mask to be saved. Color Mask files have .CMK extensions.

Note

Correct .ToleranceMode and .Smoothing values may not be saved.

Example

```
.ColorMaskSave "C:\MASK1.CMK"
```

The above example saves the color mask as a file named "MASK1.CMK".

{button ,AL(`PP_ColorMask_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.ColorReplace (PHOTO-PAINT)

This command modifies all pixels similar or the same to the current paint color in the image to the active paper color.

Syntax
.ColorReplace

Note

The .ColorReplace command must be preceded by the appropriate tool settings commands:
.SetPaintColor
.PenSettings
.ToleranceSettings

Example

```
.ColorReplace
```

The above example replaces the current paint color in your image with the paper color.

{button ,AL(`PP_Tools_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.ContinueDraw (PHOTO-PAINT)

This command continues tool commands started with the StartDraw command such as drawing, masking, erasing, and filling.

Syntax

.ContinueDraw .ptX=*long*, .ptY=*long*

Syntax	Definition
.ptX	Specifies the X coordinate of the point to continue drawing in pixels, relative to the origin.
.ptY	Specifies the Y coordinate of the point to continue drawing in pixels, relative to the origin.

Note

The .ContinueDraw command must be in a contiguous block of one or more .ContinueDraw commands. The first .ContinueDraw command in the block must be preceded by the .StartDraw command, and the last must be followed by the .EndDraw command.

The .StartDraw command must be preceded by the appropriate tool settings commands.

Example

```
.SetPaintColor 5, 153, 0, 204, 0
.PenSettings 17, 5, 0, 0, 0, 0, 0
.ShapeSettings 0, 0, 5, -1, -1
.StartDraw 162, 365
.ContinueDraw 308, 203
.ContinueDraw 76, 203
.ContinueDraw 162, 365
.EndDraw
```

The above example draws a triangle with points at coordinates (162, 365), (308, 203), and (76, 203). These coordinates are expressed in pixels.

```
.StartDraw 140, 169
.ContinueDraw 138, 168
.ContinueDraw 136, 166
.ContinueDraw 130, 160
...
...
...
.ContinueDraw 136, 167
.ContinueDraw 137, 167
.EndDraw
```

The above example creates a Lasso Mask.

{button ,AL(`PP_Draw_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.SetDocVisible (PHOTO-PAINT)

This command controls whether the document is visible in PHOTO-PAINT.

Syntax
.SetDocVisible .Show=*boolean*

Syntax	Definition
.Show	Set to TRUE (-1) to display the active document. Set to FALSE (0) to hide the active document.

Example

```
.SetDocVisible -1
```

The above example displays the active Corel PHOTO-PAINT document.

{button ,AL(`PP_View_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.EditCheckpoint (PHOTO-PAINT)

This command saves a copy of the image to a temporary file. Additions or edits to the image beyond the checkpoint can be removed by using the .EditRestoreCheckpoint command. This will reverse all previous changes made to the image since this command was selected.

Syntax
.EditCheckpoint

Note

Issuing this command clears the undo list.

Example

```
.EditCheckpoint
```

The above example saves the image at its current state.

{button ,AL(`PP_Edit_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.EditClear (PHOTO-PAINT)

If an object is selected , this command deletes the object. If a mask is present, this command clears the masked area to the current paper color. If no mask is present, this command clears the entire image to the current paper color. If an object is selected and a mask is present, the object is deleted and the mask ignored.

Syntax
.EditClear

Example
```
.EditClear
```

The above example clears the current document and removes any editing that has been performed on it.

{button ,AL(`PP_Edit_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.EditClearClipboard (PHOTO-PAINT)

This command clears the clipboard of all information. This is useful for memory conservation as well as reducing execution time.

Syntax
.EditClearClipboard

Example
```
.EditClearClipboard
```

The above example removes all items previously placed on the clipboard.

{button ,AL(`PP_Edit_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.EditCopy (PHOTO-PAINT)

This command copies an object or masked area from the image and places it on the clipboard. If there is no mask present or object selected, the entire image is copied to the clipboard.

Syntax
.EditCopy

Example
```
.EditCopy
```

The above example copies the selected object to the clipboard.

{button ,AL(`PP_Edit_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.EditCopyToFile (PHOTO-PAINT)

This command saves a copy of an object or masked area to an existing or new file. If there is no mask present or object selected, the entire image is copied to the file.

Syntax

.EditCopyToFile .FileName=*string*, .FilterID=*long*, .Compression=*long*

Syntax	Definition
.FileName	The path and file name of the destination file.
.FilterID	Specifies the type of file filter. 769 = Windows Bitmap (BMP) 770 = Paintbrush (PCX) 771 = Targa Bitmap (TGA) 772 = TIFF Bitmap (TIF) 773 = CompuServe Bitmap (GIF) 774 = JPEG Bitmaps (JPG) 775 = Kodak Photo CD Image (PCD) 776 = Scitex CT Bitmap (SCT) 777 = Wavelet Compressed Bitmap (WVL) 787 = GEM Paint File (IMG) 784 = Windows 3.x/NT Cursor Resource (CUR) 788 = Adobe Photoshop (PSD) 785 = Windows 3.x/NT Icon Resource (ICO) 786 = Windows 3.x/NT Bitmap Resource (EXE) 790 = MACPaint Bitmap (MAC) 789 = Picture Publisher 4 (PB4) 800 = CALS Compressed Bitmap (CAL) 1280 = Computer Graphics Metafile (CGM) 1281 = HPGL Plotter File (PLT) 1283 = Adobe Illustrator (AI) 1284 = GEM File (GEM) 1285 = IBM PIF (PIF) 1287 = WordPerfect Graphics (WPG) 1288 = Macintosh Pict (PCT) 1289 = Encapsulated PostScript (EPS) 1290 = PostScript Interpreted (PS) 1291 = OS/2 PM Metafile (MET) 1294 = Windows Metafile (WMF) 1295 = Corel Metafile (CMF) 1296 = AutoCad (DXF) 1536 = Video for Windows (AVI) 1539 = CorelMOVE (CMV) 1540 = CorelSHOW (SHW) 1541 = CorelCHART (CCH) 1543 = AutoDesk FLIC (FLI) 1548 = MicroSoft PowerPoint (PPT) 1549 = Lotus Freelance (PRE) 1551 = MPEG Animation (MPG) 1792 = Corel PHOTO-PAINT Image (CTP) 1794 = Corel CMX 5.0 1793 = Corel CMX 6.0 1795 = CorelDRAW (CDR)
.Compression	Specifies the type of file compression to apply: 0 = None 1 = LZW 2 = Packbits 3 = JPEG

Example

```
.EditCopyToFile "C:\COREL60\PHOTOPNT\TEST1.CPT", 1792, 0
```

The above example copies the selected object to the PHOTO-PAINT file named "TEST1.CPT".

{button ,AL(`PP_Edit_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.EditCut (PHOTO-PAINT)

This command cuts an object or masked area from the image and places it on the clipboard. If there is no mask present or object selected, the entire image is cut and placed on the clipboard.

Syntax
.EditCut

Note

This command is only available when a mask or object is active and selected in the main window.

Example

```
.EditCut
```

The above example removes the selected object from the document and places it on the clipboard.

{button ,AL(`PP_Edit_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.EditPasteDocument (PHOTO-PAINT)

This command creates a new document and inserts the contents of the clipboard as an object. The size of the new document will be the same as the pasted object. The background paper color will be the current paper color.

Syntax
.EditPasteDocument

Example
```
.EditPasteDocument
```

The above example inserts a copy of the object most recently placed in the clipboard into a new document.

{button ,AL(`PP_Edit_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.EditPasteFromFile (PHOTO-PAINT)

This command lets you select an image from a file to paste into the active image.

Syntax

.EditPasteFromFile .FileName=*string*, .Left=*long*, .Top=*long*, .Right=*long*, .Bottom=*long*, .LoadType =*long*

Syntax	Definition
.FileName	Specifies the path and name of the source file.
.Left	Specifies the X coordinate of the upper-left corner of the file's bounding rectangle in pixels, relative to the origin.
.Top	Specifies the Y coordinate of the upper-left corner of the file's bounding rectangle in pixels, relative to the origin.
.Right	Specifies the X coordinate of the lower-right corner of the file's bounding rectangle in pixels, relative to the origin.
.Bottom	Specifies the Y coordinate of the lower-right corner of the file's bounding rectangle in pixels, relative to the origin.
.LoadType	Specifies the image load type of the file: 0 = All 1 = Partial 2 = Resample 3 = Crop

Example

```
.EditPasteFromFile "C:\COREL60\PHOTOPNT\CLAMSHEL.BMP", 0, 0, 0, 0, 0
```

The above example pastes the bitmap file named "CLAMSHEL.BMP" into the open image.

{button ,AL(`PP_Edit_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.EditPasteObject (PHOTO-PAINT)

This command pastes the current clipboard contents as a new object into the active image.

Syntax
.EditPasteObject

Example
```
.EditPasteObject
```

The above example pastes a copy of the object most recently placed in the clipboard into the current document.

{button ,AL(`PP_Edit_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.EditRestoreCheckpoint (PHOTO-PAINT)

This command returns an image to the state it was at when the .EditCheckpoint command was last used.

Syntax
.EditRestoreCheckpoint

Example
```
.EditRestoreCheckpoint
```

The above example returns the active image to the state it was at when the .EditCheckpoint command was last used.

{button ,AL(`PP_Edit_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.EditUndo (PHOTO-PAINT)

This command cancels the last change that was made to the image and returns it to its former state.

Syntax
.EditUndo

Example
```
.EditUndo
```

The above example cancels the last change that was made to the image.

{button ,AL(`PP_Edit_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.Ellipse (PHOTO-PAINT)

This command draws hollow/filled ellipses.

Syntax

.Ellipse .Left=*long*, .Top=*long*, .Right=*long*, .Bottom=*long*

Syntax	Definition
.Left	Specifies the X coordinate of the upper-left corner of the ellipse's bounding rectangle in pixels, relative to the origin.
.Top	Specifies the Y coordinate of the upper-left corner of the ellipse's bounding rectangle in pixels, relative to the origin.
.Right	Specifies the X coordinate of the lower-right corner of the ellipse's bounding rectangle in pixels, relative to the origin.
.Bottom	Specifies the Y coordinate of the lower-right corner of the ellipse's bounding rectangle in pixels, relative to the origin.

Note

The .Ellipse command must be preceded by the appropriate tool settings commands:
.SetPaintColor
.PenSettings
.ShapeSettings

Example

```
.Ellipse 58, 46, 263, 130
```

The above example creates an ellipse with the upper-left corner of the ellipse's bounding rectangle at the point (58, 46) and the lower-right corner of the ellipse's bounding rectangle at the point (263, 130). These coordinates are expressed in pixels.

{button ,AL(`PP_Draw_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.EndDraw (PHOTO-PAINT)

This command ends the sequence used with drawing, masking, erasing, and filling tools.

Syntax
.EndDraw

Note

The .EndDraw command must be preceded by a .StartDraw command and a contiguous block of one or more .ContinueDraw commands.

The .StartDraw command must be preceded by the appropriate tool settings commands.

Example

```
.StartDraw 162,365
.ContinueDraw 308,203
.ContinueDraw 76,203
.ContinueDraw 162,365
.EndDraw
```

The above example draws a triangle with points at coordinates (162, 365), (308, 203), and (76, 203). These coordinates are expressed in pixels.

```
.StartDraw 140, 169
.ContinueDraw 138, 168
.ContinueDraw 136, 166
.ContinueDraw 130, 160
...
...
...
.ContinueDraw 136, 167
.ContinueDraw 137, 167
.EndDraw
```

The above example creates a Lasso Mask.

{button ,AL(`PP_Draw_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.FileClose (PHOTO-PAINT)

This command closes the active image.

Syntax
.FileClose

Example
```
.FileClose
```

The above example closes the active document.

{button ,AL(`PP_File_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.FileNew (PHOTO-PAINT)

This command defines the attributes/characteristics of a new image.

Syntax

.FileNew .Width=*long*, .Height=*long*, .Type=*long*, .HRes=*long*, .VRes=*long*, .PartialFile=*boolean*, .MovieFile=*boolean*, .NumberFrames=*long*, .Left=*long*, .Top=*long*, .Right=*long*, .Bottom=*long*, .Color1=*long*, .Color2=*long*, .Color3=*long*, .Color4=*long*

Syntax	Definition
.Width	Specifies the width of the new image in pixels.
.Height	Specifies the height of the new image in pixels.
.Type	Specifies the paint image type of the file: 1 = RGB 2 = 256 Gray 3 = Black & White 4 = 256 Color 5 = CMYK 6 = 16 Color
.HRes	Specifies the horizontal resolution of the image in dots per inch (dpi).
.VRes	Specifies the vertical resolution of the image in dots per inch (dpi).
.PartialFile	Set to TRUE (-1) to load a partial area of an image. This lets you work on separate areas of an image without opening the entire file. Otherwise set to FALSE (0). Note: If this option is selected, the new file cannot be a movie file.
.MovieFile	Set to TRUE (-1) to open the new image as a movie file. Otherwise set to FALSE (0). Note: If the new image is opened as a movie file, the partial file option is disabled.
.NumberFrames	If the movie file option is enabled (ie. set to TRUE) enter the number of frames you would like the movie file to contain.
.Left	Specifies the X coordinate of the upper-left corner of the file's bounding rectangle in pixels, relative to the origin. (Applies only if the new image is a partial file.)
.Top	Specifies the Y coordinate of the upper-left corner of the file's bounding rectangle in pixels, relative to the origin. (Applies only if the new image is a partial file.)
.Right	Specifies the X coordinate of the lower-right corner of the file's bounding rectangle in pixels, relative to the origin. (Applies only if the new image is a partial file.)
.Bottom	Specifies the Y coordinate of the lower-right corner of the file's bounding rectangle in pixels, relative to the origin. (Applies only if the new image is a partial file.)
.Color1	Specifies the first color component for .ColorModel. For example, Red is the first color component for RGB. Click + for valid value ranges.

.Color2 Specifies the second color component for .ColorModel. For example, Green is the second color component for RGB. Click + for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.
.Color3 Specifies the third color component for .ColorModel. For example, Blue is the third color component for RGB. Click + for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.

.Color4	Specifies the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. Click + for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.

Note

The .ColorModel must correspond to the selected image type.

Example

```
.FileNew 320, 200, 1,75, 75, 0, 0, 1, 0, 0, 0, 0, 211, 119, 0, 0
```

The above example creates a new document with the following attributes:
.Width = 320 pixels (4.4 inches)
.Height = 200 pixels (2.7 inches)
.Type = 24 bit color (RGB)
.HRes = 75 dpi
.VRes = 75 dpi
.PartialFile = FALSE (disabled)
.MovieFile = FALSE (disabled)

```
.FileNew 360, 504, 1, 72, 72, 0, 0, 1, 0, 0, 0, 255, 255, 255, 0
```

The above example creates a new document with the following attributes:
.Width = 360 pixels (5 inches)
.Height = 504 pixels (7 inches)
.Type = 24 bit color (RGB)
.HRes = 72 dpi
.VRes = 72 dpi
.PartialFile = FALSE (disabled)
.MovieFile = FALSE (disabled)

{button ,AL(`PP_File_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.FileOpen (PHOTO-PAINT)

This command opens an existing file and loads it into the main image window.

Syntax

.FileOpen .FileName=*string*, .Left=*long*, .Top=*long*, .Right=*long*, .Bottom=*long*, .LoadType=*long*

Syntax	Definition
.FileName	Specifies the path and name of the file to be opened.
.Left	Specifies the X coordinate of the upper-left corner of the file's bounding rectangle in pixels, relative to the origin.
.Top	Specifies the Y coordinate of the upper-left corner of the file's bounding rectangle in pixels, relative to the origin.
.Right	Specifies the X coordinate of the lower-right corner of the file's bounding rectangle in pixels, relative to the origin.
.Bottom	Specifies the Y coordinate of the lower-right corner of the file's bounding rectangle in pixels, relative to the origin.
.LoadType	Indicates the image load type of the file: 0 = All (coordinates are not used) 1 = Partial 2 = Resample 3 = Crop

Example

```
.FileOpen "C:\COREL60\CUSTOM\TEST.1.CPT", 0, 0, 0, 0, 0
```

The above example opens the Corel PHOTO-PAINT file named "TEST.1.CPT".

{button ,AL(`PP_File_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.FilePrint (PHOTO-PAINT)

This command sends the current document to the printer.

Syntax
.FilePrint

Example
```
.FilePrint
```

The above example sends the active document to the printer.

{button ,AL(`PP_File_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.FileRevert (PHOTO-PAINT)

This command undoes changes made to the image since it was last saved. Use this command when Undo is unavailable or when you want to undo more than one action.

Syntax
.FileRevert

Example
```
.FileRevert
```

The above example undoes changes made to the image since it was last saved.

{button ,AL(`PP_File_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.FileSave (PHOTO-PAINT)

This command saves the current image.

Syntax
.FileSave .FileName=*string*, .FilterID=*long*, .Compression=*long*

Syntax	Definition
.FileName	Specifies the path and name of the file to be saved.
.FilterID	Specifies the type of file filter. 769 = Windows Bitmap (BMP) 770 = Paintbrush (PCX) 771 = Targa Bitmap (TGA) 772 = TIFF Bitmap (TIF) 773 = CompuServe Bitmap (GIF) 774 = JPEG Bitmaps (JPG) 775 = Kodak Photo CD Image (PCD) 776 = Scitex CT Bitmap (SCT) 777 = Wavelet Compressed Bitmap (WVL) 787 = GEM Paint File (IMG) 784 = Windows 3.x/NT Cursor Resource (CUR) 788 = Adobe Photoshop (PSD) 785 = Windows 3.x/NT Icon Resource (ICO) 786 = Windows 3.x/NT Bitmap Resource (EXE) 790 = MACPaint Bitmap (MAC) 789 = Picture Publisher 4 (PB4) 800 = CALS Compressed Bitmap (CAL) 1280 = Computer Graphics Metafile (CGM) 1281 = HPGL Plotter File (PLT) 1283 = Adobe Illustrator (AI) 1284 = GEM File (GEM) 1285 = IBM PIF (PIF) 1287 = WordPerfect Graphics (WPG) 1288 = Macintosh Pict (PCT) 1289 = Encapsulated PostScript (EPS) 1290 = PostScript Interpreted (PS) 1291 = OS/2 PM Metafile (MET) 1294 = Windows Metafile (WMF) 1295 = Corel Metafile (CMF) 1296 = AutoCad (DXF) 1536 = Video for Windows (AVI) 1539 = CorelMOVE (CMV) 1540 = CorelSHOW (SHW) 1541 = CorelCHART (CCH) 1543 = AutoDesk FLIC (FLI) 1548 = MicroSoft PowerPoint (PPT) 1549 = Lotus Freelance (PRE) 1551 = MPEG Animation (MPG) 1792 = Corel PHOTO-PAINT Image (CTP) 1794 = Corel CMX 5.0 1793 = Corel CMX 6.0 1795 = CorelDRAW (CDR)
.Compression	Specifies the type of file compression to apply: 0 = None 1 = LZW 2 = Packbits 3 = JPEG

Example

```
.FileSave "C:\COREL60\PHOTOPNT\TEST1.BMP", 769, 0
```

The above example saves the file named "TEST1.BMP" in Windows bit map format with no compression applied.

{button ,AL(`PP_File_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.Fill (PHOTO-PAINT)

This command fills areas based on the color similarities of adjacent pixels.

Syntax

.Fill .Left=*long*, .Top=*long*, .AntiAlias=*boolean*

Syntax	Definition
.Left	Specifies the X coordinate of the point where filling begins in pixels, relative to the origin.
.Top	Specifies the Y coordinate of the point where filling begins in pixels, relative to the origin.
.AntiAlias	Set to TRUE (-1) to apply anti-aliasing. Set to FALSE (0) to disable anti-aliasing.

Note

The .Fill command must be preceded by the appropriate tool settings commands:
.PenSettings
.ToleranceSettings

Example

```
.PenSettings 24, 10, 0, 0, 0, 0, 0
.ToleranceSettings 0, 10, 10, 10, 10
.FillSolid 5, 0, 0, 255, 0
.Fill 114, 69, 0
```

The above example applies a solid blue fill beginning at the point (103, 82) with no anti-aliasing. These coordinates are expressed in pixels.

{button ,AL(`PP_Draw_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.GetColorMaskPresent (PHOTO-PAINT)

This function returns a value indicating whether there is a Color Mask present.
1 = Color Mask is present.
0 = Color Mask is not present.

Syntax
.GetColorMaskPresent

Example

```
status& = .GetColorMaskPresent()
MESSAGE status&
```

The above example determines whether a color mask is present.

{button ,AL(`PP_Get_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.GetDocumentHeight (PHOTO-PAINT)

This function returns the height of the active image in pixels.

Syntax
.GetDocumentHeight

Example

```
h& = .GetDocumentHeight()
MESSAGE h&
```

The above example gets the height of the active document.

{button ,AL(`PP_Get_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.GetDocumentIsMovie (PHOTO-PAINT)

This function returns a value indicating whether the active document is a movie file.
TRUE (-1) = Document is a movie file.
FALSE (0) = Document is not a movie file.

Syntax
.GetDocumentIsMovie

Example

```
status& = .GetDocumentIsMovie()
MESSAGE status&
```

The above example determines whether the document is a movie file.

{button ,AL(`PP_Get_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.GetDocumentIsPartial (PHOTO-PAINT)

This function returns a value indicating whether the active document is a partial file.
TRUE (-1) = Document is a partial file.
FALSE (0) = Document is not a partial file.

Syntax
.GetDocumentIsPartial

Example

```
status& = .GetDocumentIsPartial()
MESSAGE status&
```

The above example determines whether the document is a partial file.

{button ,AL(`PP_Get_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.GetDocumentName (PHOTO-PAINT)

This function returns a string containing the name and file extension of the active image.

Syntax
.GetDocumentName

Example

```
name$ = .GetDocumentName()
MESSAGE name$
```

The above example determines the name of the active document.

{button ,AL(`PP_Get_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.GetDocumentType (PHOTO-PAINT)

This function returns a value indicating the paint image type of the active document.

Syntax
.GetDocumentType

Example

```
docType$ = .GetDocumentType()
MESSAGE docType$
```

The above example determines the paint image type of the active document.

{button ,AL(`PP_Get_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.GetDocumentWidth (PHOTO-PAINT)

This function returns the width of the active image in pixels.

Syntax
.GetDocumentWidth

Example

```
w& = .GetDocumentWidth()
MESSAGE status&
```

The above example gets the width of the active document.

{button ,AL(`PP_Get_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.GetDocumentXdpi (PHOTO-PAINT)

This function returns the horizontal resolution of the active image in dots per inch (dpi).

Syntax
.GetDocumentXdpi

Example

```
xdpi& = .GetDocumentXdpi()
MESSAGE xdpi&
```

The above example gets the horizontal resolution of the active document.

{button ,AL(`PP_Get_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.GetDocumentYdpi (PHOTO-PAINT)

This function returns the vertical resolution of the active image in dots per inch (dpi).

Syntax
.GetDocumentYdpi

Example

```
ydpi& = .GetDocumentYdpi()
MESSAGE ydpi&
```

The above example gets the vertical resolution of the active document.

{button ,AL(`PP_Get_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.GetFrameCount (PHOTO-PAINT)

This function returns the number of frames in the active image.

Syntax
.GetFrameCount

Example

```
cnt& = .GetFrameCount()
MESSAGE cnt&
```

The above example determines the frame count of the active document.

{button ,AL(`PP_Get_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.GetMaskPresent (PHOTO-PAINT)

This function returns a value indicating whether there is a Mask present.
1 = Mask is present.
0 = Mask is not present.

Syntax
.GetMaskPresent

Example

```
status& = .GetMaskPresent()
MESSAGE status&
```

The above example determines whether there is a mask present.

{button ,AL(`PP_Get_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.GetObjectCount (PHOTO-PAINT)

This function returns the number of objects in the active image.

Syntax
.GetObjectCount

Note

The background is not included in the object count.

Example

```
cnt& = .GetObjectCount()
MESSAGE cnt&
```

The above example determines the number of objects in the active document.

{button ,AL(`PP_Get_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.GetObjectMergeMode (PHOTO-PAINT)

This function returns a value indicating the Merge Mode for the selected object:
0 = Normal
1 = Add
2 = Subtract
3 = XOR

Syntax
.GetObjectMergeMode .ObjectID=*long*

Syntax	Definition
.ObjectID	Identifies the object to which the Merge Mode is applied.

Example

```
mode& = .GetObjectMergeMode(2)
MESSAGE mode&
```

The above example determines the specified object's merge mode.

{button ,AL(`PP_Get_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.GetObjectName (PHOTO-PAINT)

This function returns a string containing the name of the selected object.

Syntax
.GetObjectName .ObjectID=*long*

Syntax	Definition
.ObjectID	Identifies the object.

Example

```
name$ = .GetObjectName(2)
MESSAGE name$
```

The above example determines the specified object's name.

{button ,AL(`PP_Get_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.GetObjectOpacity (PHOTO-PAINT)

This function returns a value (1 to 100) indicating the opacity of the active image. Opacity refers to the ability to see through an object.

Syntax
.GetObjectOpacity .ObjectID=*long*

Syntax	Definition
.ObjectID	Identifies the degree of opacity applied to the selected object.

Example

```
opacity& = .GetObjectOpacity(2)
MESSAGE opacity&
```

The above example determines the opacity of the specified object.

{button ,AL(`PP_Get_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.GridSetup (PHOTO-PAINT)

This command sets the grid attributes.

Syntax

.GridSetup .GridX=*double*, .GridY=*double*, .Units=*long*, .GridOn=*boolean*

Syntax	Definition
.GridX	Specifies the horizontal spacing between units on the grid in the user's selected units.
.GridY	Specifies the horizontal spacing between units on the grid in the user's selected units.
.Units	Specifies the user's selected unit: 1 = inch 2 = millimeters 3 = pica points 4 = fractional points 5 = centimeters 6 = pixels 12 = ciceros 13 = didots
.GridOn	Set to TRUE (-1) to turn the Grid on. Set to FALSE (0) to turn the Grid off.

Example

```
.GridSetup 10, 10, 6, -1
```

The above example sets the horizontal and vertical spacing to 10, the units are set to pixels, and the grid is enabled.

{button ,AL(`PP_View_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.ImageColorCrop (PHOTO-PAINT)

This command crops out a specific color border surrounding an image to the point where a different colored pixel is encountered. The cropping produces a new image with as much of the border removed without affecting the principle image.

Syntax

.ImageColorCrop .ColorModel=*long*, .Color1=*long*, .Color2=*long*, .Color3=*long*, .Color4=*long*, .ToleranceMode=*long*, .Normal=*long*, .Hue=*long*, .Saturation=*long*, .Brightness=*long*

Syntax	Definition
.ColorModel	Specifies the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Specifies the first color component for .ColorModel. For example, Red is the first color component for RGB. Click [+] for valid value ranges.

.Color2 Specifies the second color component for .ColorModel. For example, Green is the second color component for RGB. Click [+] for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.

.Color3	Specifies the third color component for .ColorModel. For example, Blue is the third color component for RGB. Click [+] for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.

.Color4 Specifies the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. Click [+] for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.

.ToleranceMode	Specifies the mode used to crop the image: 0 = Normal 1 = HSB Note: If Normal mode is selected, set Hue, Saturation, and Brightness to 0.
.Normal	Set to 0 to enable normal mode. Set to 1 to enable HSB mode. Note: If Normal mode is selected, set Hue, Saturation, and Brightness to 0.
.Hue	Specifies the hue. In the HSB color model, hue is the main attribute in a color that distinguishes it from other colors. Blue, green and red, for example, are all hues.
.Saturation	Specifies the saturation. Saturation is the purity of a color. The HSB color model uses Saturation as a component that determines the purity or intensity of a color. The more colors used to mix a color, the duller the color looks.
.Brightness	Specifies the brightness. In the HSB color model, the component that determines the amount of black in a color.

Example

```
.ImageColorCrop 5, 0, 0, 204, 0, 0, 0, 0, 0
```

The above example crops out a blue color border.

{button ,AL(`PP_Image_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.ImageConvert (PHOTO-PAINT)

This command converts the loaded image to another graphic format.

Syntax

.ImageConvert .Type=*long*, .RenderType=*long*, .PaletteType=*long*, .Threshold=*short*, .HalftoneType=*long*, .Angle=*long*, .Width=*long*

Syntax	Definition
.Type	Specifies the paint image type of the file: 1 = RGB 2 = 256 Gray 3 = Black & White 4 = 256 Color 5 = CMYK 6 = 16 Color 7 = Duotone
.RenderType	Specifies the type of Rendering to apply: 0 = Invalid 1 = None 2 = Error Diffusion - 2D 3 = Bayer Dithering 4 = Halftone - 4x4 5 = Halftone - 8x8 6 = Windows Dithering
.PaletteType	Specifies the Palette type used. 100 = Image 101 = Adaptive 102 = Linear 103 = Custom
.Threshold	Use the Threshold filter to gradually darken an image. RGB component values below the threshold will become 0. Component values above the threshold are not altered. For grayscale images, pixels below the threshold value become black, lighter shades remain unaltered.Valid threshold levels range from 1 to 255.
.HalftoneType	Specifies the Halftone types of the image: 0 = None 1 = Square 2 = Round 3 = Line 4 = Cross
.Angle	Specifies the angle of conversion, in degrees, to apply to the image. Positive numbers result in counter-clockwise rotation, negative numbers result in clockwise rotation.
.Width	Specifies the halftone width in pixels.

Note

You can use the ANGLECONVERT function to specify angle measurements.

Example

```
.ImageConvert 2, 1, 0, 0, 0, 0, 0
```

The above example converts the loaded image from 256 Gray Color mode to RGB Color mode.

{button ,AL(`PP_Image_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.ImageCrop (PHOTO-PAINT)

This command crops the current image to the specified rectangle.

Syntax

.ImageCrop .Left=*long*, .Top=*long*, .Right=*long*, .Bottom=*long*

Syntax	Definition
.Left	Specifies the X coordinate of the upper-left corner of the image's bounding rectangle in pixels, relative to the origin.
.Top	Specifies the Y coordinate of the upper-left corner of the image's bounding rectangle in pixels, relative to the origin.
.Right	Specifies the X coordinate of the lower-right corner of the image's bounding rectangle in pixels, relative to the origin.
.Bottom	Specifies the Y coordinate of the lower-right corner of the image's bounding rectangle in pixels, relative to the origin.

Example

```
.ImageCrop 41, 154, 173, 75
```

The above example crops the image to the specified rectangle.

{button ,AL(`PP_Image_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.ImageCropToMask (PHOTO-PAINT)

This command crops the selected image to the bounding rectangle of the current Mask.

Syntax
.ImageCropToMask

Note

A Mask must be present before using this command.

Example

```
.ImageCropToMask
```

The above example crops the selected image to the size of the current Mask.

{button ,AL(`PP_Image_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.ImageDeskew (PHOTO-PAINT)

This command adjusts a skewed or imperfectly positioned image squarely on the screen. This command is especially valuable when you are working with scanned images.

Syntax

.ImageDeskew .ColorModel=*long*, .Color1=*long*, .Color2=*long*, .Color3=*long*, .Color4=*long*

Syntax	Definition
.ColorModel	Specifies the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Specifies the first color component for .ColorModel. For example, Red is the first color component for RGB. Click ⊞ for valid value ranges.

.Color2 Specifies the second color component for .ColorModel. For example, Green is the second color component for RGB. Click ⊞ for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.

.Color3	Specifies the third color component for .ColorModel. For example, Blue is the third color component for RGB. Click ⊞ for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.

.Color4 Specifies the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. Click ⊞ for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.

Example

```
.ImageDeskew 5, 0, 119, 211,0
```

The above example uses the CMYK Color mode with Magenta and Yellow colors being applied.

{button ,AL(`PP_Image_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.ImageDuplicate (PHOTO-PAINT)

This command creates a copy of the current image and assigns it the specified name.

Syntax
.ImageDuplicate .FileName=*string*, .MergeObjects=*boolean*

Syntax	Definition
.FileName	Specifies the name and path of the original file. This is the file that will be duplicated.
.MergeObjects	Set to TRUE (-1) to enable objects to be merged. Set to FALSE (0) to disable merging of objects.

Example

```
.ImageDuplicate "NEW2.CPT", FALSE
```

The above example duplicates the file named "NEW2.CPT", without merging all objects in the duplicated image.

{button ,AL(`PP_Image_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.ImageFlipHorizontal (PHOTO-PAINT)

This command flips the image horizontally.

Syntax
.ImageFlipHorizontal

Example
```
.ImageFlipHorizontal
```

The above example flips the active image horizontally.

{button ,AL(`PP_Image_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.ImageFlipVertical (PHOTO-PAINT)

This command flips the image vertically.

Syntax
.ImageFlipVertical

Example
```
.ImageFlipVertical
```

The above example flips the active image vertically.

{button ,AL(`PP_Image_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.ImagePapersize (PHOTO-PAINT)

This command adjusts/changes the size of the paper behind an image using absolute width and height values. The image can be repositioned on the paper by setting a placement position.

Syntax

.ImagePapersize .Width=*long*, .Height=*long*, .Xoffset=*long*, .Yoffset=*long*

Syntax	Definition
.Width	Specifies the horizontal paper width in pixels.
.Height	Specifies the vertical paper height in pixels.
.Xoffset	Specifies the placement of the image on the paper offset from the x-axis (expressed in pixels).
.Yoffset	Specifies the placement of the image on the paper offset from the y-axis (expressed in pixels).

Example

```
.ImagePapersize 637, 825, 0, 0
```

The above example sets the paper size to 408 pixels (width) by 528 pixels (height) with no offset applied. This is a standard 8.5 x 11 inch size paper.

{button ,AL(`PP_Image_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.ImageResample (PHOTO-PAINT)

This command adjusts/changes the dimensions and resolution of an image.

Syntax

.ImageResample .Width=*long*, .Height=*long*, .Hres=*long*, .VRes=*long*, .AntiAlias=*boolean*

Syntax	Definition
.Width	Specifies the new width of the image in pixels.
.Height	Specifies the new height of the image in pixels.
.Hres	Specifies the new horizontal resolution of the image in dots per inch (dpi).
.VRes	Specifies the new vertical resolution of the image in dots per inch (dpi).
.AntiAlias	Set to TRUE (-1) to apply anti-aliasing. Set to FALSE (0) to disable anti-aliasing.

Example

```
.ImageResample 410, 256, 75, 75, -1
```

The above example displays the original image with a width of 410 pixels, a height of 256 pixels, horizontal and vertical resolutions of 75 dpi, applying anti-aliasing.

{button ,AL(`PP_Image_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.ImageRotate (PHOTO-PAINT)

This command rotates the selected image the specified angle and direction.

Syntax

.ImageRotate .Angle=*long*, .Clip=*boolean*, .AntiAlias=*boolean*

Syntax	Definition
.Angle	Specifies the degree of image rotation expressed in degrees. Positive numbers result in counter-clockwise rotation, negative numbers result in clockwise rotation.
.Clip	Set to TRUE (-1) to clip the image to original size. Set to FALSE (0) to increase the image size.
.AntiAlias	Set to TRUE (-1) to apply anti-aliasing. Set to FALSE (0) to disable anti-aliasing.

Note

You can use the ANGLECONVERT function to specify angle measurements.

Example

```
.ImageRotate 315, -1, 0
```

The above example rotates the image 45 degrees clockwise, maintaining the original size without applying anti-aliasing.

```
.ImageRotate 45, -1, -1
```

The above example rotates the image 45 degrees counter-clockwise, maintaining the original size and applying anti-aliasing.

{button ,AL(`PP_Image_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.ImageSplit (PHOTO-PAINT)

This command separates an image into the color channels corresponding to the specified/selected color model.

Syntax
.ImageSplit .Type=*long*

Syntax	Definition
.Type	Specifies the Image Type: 0 = RGB - split to all types 1 = CMYK - split only to CMYK channels 2 = HSB - cannot be split 3 = HLS - cannot be split 4 = YIQ - cannot be split

Example

```
.ImageSplit 1
```

The above example splits the current frame into the 4 CMYK channels: Cyan, Magenta, Yellow, and Black.

{button ,AL(`PP_Image_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.MaskAntiAlias (PHOTO-PAINT)

This command controls the degree of anti-aliasing on the selected mask. Anti-aliasing creates a smooth mask by removing jagged edges from the original mask and by adding duplicated pixels where the mask edge contacts the background.

Syntax
.MaskAntiAlias .Tolerance=*long*

Syntax	Definition
.Tolerance	Specifies the degree of anti-aliasing on the selected mask. Tolerance values range from 1 to 100.

Example

```
.MaskAntiAlias 20
```

The above examples provides anti-aliasing of 20.

{button ,AL(`PP_Mask_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.MaskAverage (PHOTO-PAINT)

This command calculates the average number of pixels around the edge of the mask and creates a smooth transition between the mask and background.

Syntax

.MaskAverage .Radius=*long*

Syntax	Definition
.Radius	Specifies the radius, in pixels, of the averaging area.

Example

```
.MaskAverage 100
```

The above example uses an averaging area of 100 pixels to create a smooth transition between the mask and the background.

{button ,AL(`PP_Mask_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.MaskBorder (PHOTO-PAINT)

This command creates a duplicate mask which surrounds and mirrors the original, creating a mask border.

Syntax
.MaskBorder .Width=*long*, .Edges=*long*

Syntax	Definition
.Width	Specifies the width of the Mask border, in pixels.
.Edges	Specifies the type of feather: 0 = Soft 1 = Medium 3 = Hard

Example

```
.MaskBorder 10, 0
```

The above example creates a mask border with a width of 10 pixels and a soft border edge.

{button ,AL(`PP_Mask_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.MaskChannelAdd (PHOTO-PAINT)

This command saves a mask as a Mask Channel.

Syntax
.MaskChannelAdd .MaskName=*string*

Syntax	Definition
.MaskName	Specifies the Mask Channel name to save.

Example
```
.MaskChannelAdd "Mask1"
```

The above example saves the current mask as a channel named Mask1 .

{button ,AL(`PP_MaskChannel_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.MaskChannelDelete (PHOTO-PAINT)

This command deletes the specified Mask Channel.

Syntax
.MaskChannelDelete .MaskID=*long*

Syntax	Definition
.MaskID	Identifies the Mask Channel to delete.

Example
```
.MaskChannelDelete 5
```

The above example deletes the specified mask channel.

{button ,AL(`PP_MaskChannel_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.MaskChannelLoad (PHOTO-PAINT)

This command loads a mask as a Mask Channel.

Syntax

.MaskChannelLoad .FileName=*string*, .Left=*long*, .Top=*long*, .Right=*long*, .Bottom=*long*, .LoadType=*long*

Syntax	Definition
.FileName	Specifies the file name containing the mask to load.
.Left	Specifies the X coordinate of the upper-left corner of the mask in pixels, relative to the origin.
.Top	Specifies the Y coordinate of the upper-left corner of the mask in pixels, relative to the origin.
.Right	Specifies the X coordinate of the lower-right corner of the mask in pixels, relative to the origin.
.Bottom	Specifies the Y coordinate of the upper-left corner of the mask in pixels, relative to the origin.
.LoadType	Specifies the image load type of the file: 0 = All 1 = Partial 2 = Resample 3 = Crop

Example

```
.MaskChannelLoad "C:\COREL60\PHOTOPNT\PIXELATE.PCX", 0, 0, 0, 0, 0
```

The above example loads the mask channel named "PIXELATE.CPT".

{button ,AL(`PP_MaskChannel_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.MaskChannelName (PHOTO-PAINT)

This command sets the name of an existing mask.

Syntax
.MaskChannelName .MaskID=*long*, .Name=*string*

Syntax	Definition
.MaskID	Identifies the Mask Channel.
.Name	Specifies the new name of the Mask Channel.

Example

```
.MaskChannelName 5, "Mask1"
```

The above example names the specified mask channel.

{button ,AL(`PP_MaskChannel_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.MaskChannelSave (PHOTO-PAINT)

This command saves the mask in the specified channel to a file.

Syntax
.MaskChannelSave .MaskID=*long*, .FileName=*string*, .FilterID=*long*, .Compression=*long*

Syntax	Definition
.MaskID	Identifies the Mask Channel.
.FileName	Specifies the name of the file.
.FilterID	Specifies the type of file filter. 769 = Windows Bitmap (BMP) 770 = Paintbrush (PCX) 771 = Targa Bitmap (TGA) 772 = TIFF Bitmap (TIF) 773 = CompuServe Bitmap (GIF) 774 = JPEG Bitmaps (JPG) 775 = Kodak Photo CD Image (PCD) 776 = Scitex CT Bitmap (SCT) 777 = Wavelet Compressed Bitmap (WVL) 787 = GEM Paint File (IMG) 784 = Windows 3.x/NT Cursor Resource (CUR) 788 = Adobe Photoshop (PSD) 785 = Windows 3.x/NT Icon Resource (ICO) 786 = Windows 3.x/NT Bitmap Resource (EXE) 790 = MACPaint Bitmap (MAC) 789 = Picture Publisher 4 (PB4) 800 = CALS Compressed Bitmap (CAL) 1280 = Computer Graphics Metafile (CGM) 1281 = HPGL Plotter File (PLT) 1283 = Adobe Illustrator (AI) 1284 = GEM File (GEM) 1285 = IBM PIF (PIF) 1287 = WordPerfect Graphics (WPG) 1288 = Macintosh Pict (PCT) 1289 = Encapsulated PostScript (EPS) 1290 = PostScript Interpreted (PS) 1291 = OS/2 PM Metafile (MET) 1294 = Windows Metafile (WMF) 1295 = Corel Metafile (CMF) 1296 = AutoCad (DXF) 1536 = Video for Windows (AVI) 1539 = CorelMOVE (CMV) 1540 = CorelSHOW (SHW) 1541 = CorelCHART (CCH) 1543 = AutoDesk FLIC (FLI) 1548 = MicroSoft PowerPoint (PPT) 1549 = Lotus Freelance (PRE) 1551 = MPEG Animation (MPG) 1792 = Corel PHOTO-PAINT Image (CTP) 1794 = Corel CMX 5.0 1793 = Corel CMX 6.0 1795 = CorelDRAW (CDR)
.Compression	Specifies the type of file compression to apply: 0 = None 1 = LZW 2 = Packbits 3 = JPEG

Example

```
.MaskChannelSave 0, "C:\COREL60\PHOTOPNT\MASK2.CPT", 1792, 0
```

The above example saves the mask as "MASK2.CPT".

{button ,AL(`PP_MaskChannel_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.MaskChannelToMask (PHOTO-PAINT)

This command makes the mask in the specified channel the current mask.

Syntax
.MaskChannelToMask .MaskID=*long*, .DrawMode=*long*

Syntax	Definition
.MaskID	Identifies the Mask Channel.
.DrawMode	Specifies the draw mode. 0 = Normal 1 = Add 2 = Subtract 3 = XOR

Example

```
.MaskChannelToMask 0, 33275
```

The above example makes the mask in the specified channel the current mask.

{button ,AL(`PP_MaskChannel_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.MaskRemove (PHOTO-PAINT)

This command deletes the current mask .

Syntax
.MaskRemove

Example
```
.MaskRemove
```

The above example deletes the current mask.

{button ,AL(`PP_Mask_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.MaskCreate (PHOTO-PAINT)

This command creates a mask from the selected object(s).

Syntax

.MaskCreate .PreserveImage=*boolean*, .DrawMode=*long*

Syntax	Definition
.PreserveImage	Set to TRUE (-1) to preserve the image. Set to FALSE (0) to disable this option.
.DrawMode	Specifies the draw mode. 0 = Normal 1 = Add 2 = Subtract 3 = XOR

Example

```
.MaskCreate -1, 0
```

The above example creates a mask from the selected object, preserving the image and using normal draw mode.

{button ,AL(`PP_Mask_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.MaskDeFloat (PHOTO-PAINT)

This command is used after the MaskFloaterMoveTo command to clear the current mask.

Syntax
.MaskDeFloat

Example
```
.MaskDeFloat
```

The above example creates multiple duplicates of a floating mask as you move the mask in the image area.

{button ,AL(`PP_Mask_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.MaskDistort (PHOTO-PAINT)

This command distorts the shape of the current mask.

Syntax

.MaskDistort .Corner1X=*long*, .Corner1Y=*long*, .Corner2X=*long*, .Corner2Y=*long*, .Corner3X=*long*, .Corner3Y=*long*, .Corner4X=*long*, .Corner4Y=*long*, .AntiAlias=*boolean*

Syntax	Definition
.Corner1X	Specifies the X coordinate of the new location of the upper-left corner of the mask's original bounding rectangle in pixels, relative to the origin.
.Corner1Y	Specifies the Y coordinate of the new location of the upper-left corner of the mask's original bounding rectangle in pixels, relative to the origin.
.Corner2X	Specifies the X coordinate of the new location of the upper-right corner of the mask's original bounding rectangle in pixels, relative to the origin.
.Corner2Y	Specifies the Y coordinate of the new location of the upper-right corner of the mask's original bounding rectangle in pixels, relative to the origin.
.Corner3X	Specifies the X coordinate of the new location of the lower-right corner of the mask's original bounding rectangle in pixels, relative to the origin.
.Corner3Y	Specifies the Y coordinate of the new location of the lower-right corner of the mask's original bounding rectangle in pixels, relative to the origin.
.Corner4X	Specifies the X coordinate of the new location of the lower-left corner of the mask's original bounding rectangle in pixels, relative to the origin.
.Corner4Y	Specifies the Y coordinate of the new location of the lower-left corner of the mask's original bounding rectangle in pixels, relative to the origin.
.AntiAlias	Set to TRUE (-1) to apply anti-aliasing. Set to FALSE (0) to disable anti-aliasing.

Example

```
.MaskDistort 100, 100, 100, 200, 300, 450, 200, 200
```

The above example distorts the selected mask by the specified parameters.

{button ,AL(`PP_Mask_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.MaskEllipse (PHOTO-PAINT)

This command creates elliptical or circular masks.

Syntax

.MaskEllipse .Left=*long*, .Top=*long*, .Right=*long*, .Bottom=*long*, .DrawMode=*long*, .Feather=*long*, .AntiAlias=*boolean*

Syntax	Definition
.Left	Specifies the X coordinate of the upper-left corner of the ellipse's mask in pixels, relative to the origin.
.Top	Specifies the Y coordinate of the upper-left corner of the ellipse's mask in pixels, relative to the origin.
.Right	Specifies the X coordinate of the lower-right corner of the ellipse's mask in pixels, relative to the origin.
.Bottom	Specifies the Y coordinate of the lower-right corner of the ellipse's mask in pixels, relative to the origin.
.DrawMode	Specifies the draw mode. 0 = Normal 1 = Add 2 = Subtract 3 = XOR
.Feather	Specifies the feathering width in pixels. Valid values range from 0 to 100.
.AntiAlias	Set to TRUE (-1) to apply anti-aliasing. Set to FALSE (0) to disable anti-aliasing.

Example

```
.MaskEllipse 61, 444, 314, 203, 0, 0, 0
```

The above examples creates an elliptical mask with the given coordinates, applying linear feathering, and no anti-aliasing.

{button ,AL(`PP_Tools_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.MaskExpand (PHOTO-PAINT)

This command stretches the mask by the specified amount in each direction.

Syntax
.MaskExpand .Width=*long*

Syntax	Definition
.Width	Specifies the amount of stretching in pixels. Valid values range from 1 to 200.

Example
```
.MaskExpand 15
```

The above example expands the width of the mask by 15 pixels.

{button ,AL(`PP_Mask_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.MaskFeather (PHOTO-PAINT)

This command feathers/smooths the edges of the mask.

Syntax

.MaskFeather .Width=*long*, .Direction=*long*, .Type=*long*

Syntax	Definition
.Width	Controls the number of pixels that are used in the feathering transition. A large number produces a wide feathering transitional area.
.Direction	Controls the direction of feathering. 0 = Inside 1 = Middle 2 = Outside
.Type	Specifies the type of feathering edge: 0 = Hard 1 = Soft

Example

```
.MaskFeather 10, 0, 0
```

The above examples uses 10 pixels in the feathering transition, an inside edge transition between a mask and the background, and a hard feather edge.

{button ,AL(`PP_Mask_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.MaskFlipHorizontal (PHOTO-PAINT)

This command flips the active mask horizontally.

Syntax
.MaskFlipHorizontal

Example
```
.ImageFlipHorizontal
```
The above example flips the current masked image horizontally.

{button ,AL(`PP_Mask_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.MaskFlipVertical (PHOTO-PAINT)

This command flips the active mask vertically.

Syntax
.MaskFlipVertical

Example
```
.ImageFlipVertical
```

The above example flips the current masked image vertically.

{button ,AL(`PP_Mask_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.MaskFloaterMoveTo (PHOTO-PAINT)

This command repositions the active mask.

Syntax

.MaskFloaterMoveTo .Left=*long*, .Bottom=*long*, .Copy=*boolean*

Syntax	Definition
.Left	Specifies the new X coordinate of the lower-left corner of the mask's bounding rectangle, in pixels.
.Bottom	Specifies the new Y coordinate of the lower-left corner of the mask's bounding rectangle, in pixels.
.Copy	Set to TRUE (-1) to make a copy of the image inside the mask. Set to FALSE (0) to move the image in the mask.

Example

```
.MaskFloaterMoveTo 46, 30, 0
```

The above example moves the mask up 30 pixels from the bottom edge and 46 pixels to the right from the left edge.

{button ,AL(`PP_Mask_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.MaskGrow (PHOTO-PAINT)

This command expands a mask to include areas of the image with similar pixel colors. The mask continues to expand until all of the adjacent colors that meet the selection criteria are included.

Syntax
.MaskGrow

Note

This command uses the current tolerance settings.

Example

```
.MaskGrow
```

The above example expands the mask to include areas of the image with similar pixel colors.

{button ,AL(`PP_Mask_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.MaskInvert (PHOTO-PAINT)

This command reverses the area included in the mask.

Syntax
.MaskInvert

Example
```
.MaskInvert
```

The above example inverts the current mask.

{button ,AL(`PP_Mask_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.MaskLoad (PHOTO-PAINT)

This command loads a previously saved mask or any importable image. PHOTO-PAINT converts the imported image to a grayscale mask.

Syntax

.MaskLoad .Filename=*string*, .Left=*long*, .Top=*long*, .Right=*long*, .Bottom=*long*, .LoadType=*long*

Syntax	Definition
.Filename	Specifies the name of the file.
.Left	Specifies the X coordinate of the upper-left corner of the mask to be loaded in pixels, relative to the origin.
.Top	Specifies the Y coordinate of the upper-left corner of the mask to be loaded in pixels, relative to the origin.
.Right	Specifies the X coordinate of the lower-right corner of the mask to be loaded in pixels, relative to the origin.
.Bottom	Specifies the Y coordinate of the lower-right corner of the mask to be loaded in pixels, relative to the origin.
.LoadType	Specifies the image load type of the file: 0 = All 2 = Resample 3 = Crop

Note

Mask or image cannot be loaded as a partial file.

Example

```
.MaskLoad "C:\COREL60\PHOTOPNT\MASK1.CMK", 0, 0, 0, 0, 0
```

The above example loads all of the mask named "MASK1.CPT" in to the top left corner of the document.

{button ,AL(`PP_Mask_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.MaskMagicWand (PHOTO-PAINT)

This command defines masks based on the color similarities of adjacent pixels.

Syntax

.MaskMagicWand .ptX=*long*, .ptY=*long*, .DrawMode=*long*, .AntiAlias=*boolean*

Syntax	Definition
.ptX	Specifies the X coordinate of the position at which the Magic Wand toll is applied, in pixels, relative to the origin.
.ptY	Specifies the Y coordinate of the position at which the Magic Wand toll is applied, in pixels, relative to the origin.
.DrawMode	Specifies the draw mode. 0 = Normal 1 = Add 2 = Subtract 3 = XOR
.AntiAlias	Set to TRUE (-1) to apply anti-aliasing. Set to FALSE (0) to disable anti-aliasing.

Example

```
.MaskMagicWand 10, 490, -1
```

The above example creates a magic wand mask at the point (10, 490), applying anti-aliasing. The coordinates are expressed in pixels.

{button ,AL(`PP_Tools_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.MaskPaint (PHOTO-PAINT)

This command displays a grayscale copy of the currently selected mask, allowing you to switch between editing an image or its mask.

Syntax

.MaskPaint .Mask=*long*

Syntax	Definition
.Mask	Specifies whether to paint on the mask or its image. 1 = Paint on the mask 2 = Paint on the image

Example

```
.MaskPaint 1
```

The above example paints on the mask.

{button ,AL(`PP_Mask_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.MaskRectangle (PHOTO-PAINT)

This command defines rectangular masks.

Syntax

.MaskRectangle .Left=*long*, .Top=*long*, .Right=*long*, .Bottom=*long*, .DrawMode=*long*, .Feather=*long*

Syntax	Definition
.Left	Specifies the X coordinate of the upper-left corner of the mask's rectangle in pixels, relative to the origin.
.Top	Specifies the Y coordinate of the upper-left corner of the mask's rectangle in pixels, relative to the origin.
.Right	Specifies the X coordinate of the lower-right corner of the mask's rectangle in pixels, relative to the origin.
.Bottom	Specifies the Y coordinate of the lower-right corner of the mask's rectangle in pixels, relative to the origin.
.DrawMode	Specifies the draw mode. 0 = Normal 1 = Add 2 = Subtract 3 = XOR
.Feather	Specifies the feathering width in pixels. Valid values range from 0 to 100.

Example

```
.MaskRectangle 59, 175, 210, 91, 0, 0
```

The above example creates a rectangular mask with the upper-left corner at the point (59, 175) and the lower-right corner at the point (210, 91). These coordinates are expressed in pixels.

{button ,AL(`PP_Tools_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.MaskReduce (PHOTO-PAINT)

This command reduces the size of the mask by the specified amount in each direction.

Syntax
.MaskReduce .Width=*long*

Syntax	Definition
.Width	Specifies the amount by which the bounding rectangle is inset, in pixels. Valid values range from 1 to 200.

Example

```
.MaskReduce 50
```

The above example reduces the width of the mask by 50 pixels.

```
.MaskReduce 60
```

The above example reduces the width of the mask by 60 pixels.

{button ,AL(`PP_Mask_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.MaskRemoveHoles (PHOTO-PAINT)

This command removes holes in the mask.

Syntax
.MaskRemoveHoles

Example
```
.MaskRemoveHoles
```

The above example removes holes created by masking tools.

{button ,AL(`PP_Mask_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.MaskRotate (PHOTO-PAINT)

This command rotates the current mask by the specified angle, about the given rotation point.

Syntax

.MaskRotate .XCenter=*long*, .YCenter=*long*, .Angle=*long*, .AntiAlias=*boolean*

Syntax	Definition
.XCenter	Specifies the X coordinate of the center of rotation in pixels, relative to the origin.
.YCenter	Specifies the Y coordinate of the center of rotation in pixels, relative to the origin.
.Angle	Specifies the angle of the mask's rotation in tenths of degrees. Positive numbers result in counter-clockwise rotation, negative numbers result in clockwise rotation. eg. 45 degrees clockwise = -450
.AntiAlias	Set to TRUE (-1) to apply anti-aliasing. Set to FALSE (0) to disable anti-aliasing.

Note

You can use the ANGLECONVERT function to specify angle measurements.

Example

```
.MaskRotate 268, 363, 450, 0
```

The above example rotates the mask 45 degrees around the center point located at x,y coordinates 268, 363.

{button ,AL(`PP_Mask_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.MaskSave (PHOTO-PAINT)

This command saves the active mask to a file.

Syntax
.MaskSave .Filename=*string*, .FilterID=*long*, .Compression=*long*

Syntax	Definition
.Filename	Specifies the path and name of the file.
.FilterID	Specifies the type of file filter. 769 = Windows Bitmap (BMP) 770 = Paintbrush (PCX) 771 = Targa Bitmap (TGA) 772 = TIFF Bitmap (TIF) 773 = CompuServe Bitmap (GIF) 774 = JPEG Bitmaps (JPG) 775 = Kodak Photo CD Image (PCD) 776 = Scitex CT Bitmap (SCT) 777 = Wavelet Compressed Bitmap (WVL) 787 = GEM Paint File (IMG) 784 = Windows 3.x/NT Cursor Resource (CUR) 788 = Adobe Photoshop (PSD) 785 = Windows 3.x/NT Icon Resource (ICO) 786 = Windows 3.x/NT Bitmap Resource (EXE) 790 = MACPaint Bitmap (MAC) 789 = Picture Publisher 4 (PB4) 800 = CALS Compressed Bitmap (CAL) 1280 = Computer Graphics Metafile (CGM) 1281 = HPGL Plotter File (PLT) 1283 = Adobe Illustrator (AI) 1284 = GEM File (GEM) 1285 = IBM PIF (PIF) 1287 = WordPerfect Graphics (WPG) 1288 = Macintosh Pict (PCT) 1289 = Encapsulated PostScript (EPS) 1290 = PostScript Interpreted (PS) 1291 = OS/2 PM Metafile (MET) 1294 = Windows Metafile (WMF) 1295 = Corel Metafile (CMF) 1296 = AutoCad (DXF) 1536 = Video for Windows (AVI) 1539 = CorelMOVE (CMV) 1540 = CorelSHOW (SHW) 1541 = CorelCHART (CCH) 1543 = AutoDesk FLIC (FLI) 1548 = MicroSoft PowerPoint (PPT) 1549 = Lotus Freelance (PRE) 1551 = MPEG Animation (MPG) 1792 = Corel PHOTO-PAINT Image (CTP) 1794 = Corel CMX 5.0 1793 = Corel CMX 6.0 1795 = CorelDRAW (CDR)
.Compression	Specifies the type of file compression to apply: 0 = None 1 = LZW 2 = Packbits 3 = JPEG

Example

```
.MaskSave "C:\COREL60\PHOTOPNT\MASK1.CPT", 1792, 0
```

The above example saves the current mask as a file named "MASK1.CPT".

{button ,AL(`PP_Mask_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.MaskSelectAll (PHOTO-PAINT)

This command creates a mask including the entire image area.

Syntax
.MaskSelectAll

Example
```
.MaskSelectAll
```

The above example encompasses the entire active image in a mask.

{button ,AL(`PP_Mask_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.MaskSimilar (PHOTO-PAINT)

This command creates a mask based on colors similar to those in the active mask.

Syntax
.MaskSimilar

Example
```
.MaskSimilar
```

The above example creates a mask similar to the active mask.

{button ,AL(`PP_Mask_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.MaskSkew (PHOTO-PAINT)

This command skews the current mask.

Syntax

.MaskSkew .ptX=*long*, .ptY=*long*, .HorzAngle=*long*, .VertAngle=*long*, .AntiAlias=*boolean*

Syntax	Definition
.ptX	Specifies the X coordinate of the lower-left corner of the skewed mask.
.ptY	Specifies the Y coordinate of the lower-left corner of the skewed mask.
.HorzAngle	Specifies the horizontal skew angle in tenths of degrees. Positive numbers result in counter-clockwise skew, negative numbers result in clockwise skew.
.VertAngle	Specifies the vertical skew angle tenths of degrees. Positive numbers result in counter-clockwise skew, negative numbers result in clockwise skew.
.AntiAlias	Set to TRUE (-1) to apply anti-aliasing. Set to FALSE (0) to disable anti-aliasing.

Note

You can use the ANGLECONVERT function to specify angle measurements.

Example

```
.MaskSkew 60, 120, 100, 300, 0
```

The above example skews the mask by 60 degrees horizontally and 120 degrees vertically. The X coordinate of the mask's skew is 100, and the Y coordinate is 300.

{button ,AL(`PP_Mask_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.MaskSmooth (PHOTO-PAINT)

This command smooths over or rounds off the sharp angles of a mask.

Syntax
.MaskSmooth .Radius=*long*

Syntax	Definition
.Radius	Specifies the radius of mask edge smoothing. Smoothing tones down differences in adjacent pixels along the edges of the mask. Valid values range from 1 to 50.

Example

```
.MaskSmooth 5
```

The above example applies a smoothing radius of 5 to the mask.

{button ,AL(`PP_Mask_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.MaskStretch (PHOTO-PAINT)

This command stretches the active mask.

Syntax

.MaskStretch .Left=*long*, .Top=*long*, .Right=*long*, .Bottom=*long*, .AntiAlias=*boolean*

Syntax	Definition
.Left	Specifies the X coordinate of the new upper-left corner of the stretched mask in pixels, relative to the origin.
.Top	Specifies the Y coordinate of the new upper-left corner of the stretched mask in pixels, relative to the origin.
.Right	Specifies the X coordinate of the new lower-right corner of the stretched mask in pixels, relative to the origin.
.Bottom	Specifies the Y coordinate of the new lower-right corner of the stretched mask in pixels, relative to the origin.
.AntiAlias	Set to TRUE (-1) to apply anti-aliasing. Set to FALSE (0) to disable anti-aliasing.

Example

```
.MaskStretch 100, 478, 436, 244, 0
```

The above example stretches the mask to the new coordinates shown.

{button ,AL(`PP_Mask_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.MaskStroke (PHOTO-PAINT)

This command strokes the border of the active mask using any of the brush, effect, or pen tools.

Syntax
.MaskStroke

Example
```
.MaskStroke
```

The above example strokes the border of the active mask using the current brush, effect, or pen tools.

{button ,AL(`PP_Mask_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.MaskThreshold (PHOTO-PAINT)

This command converts grayscale to a black and white (or bi-level) mask using the specified threshold value.

Syntax
.MaskThreshold .Level=*long*

Syntax	Definition
.Level	Specifies the threshold level. Values above the threshold are set to white, values below are set to black. Valid threshold levels range from 1 to 255.

Example

```
.MaskThreshold 128
```

The above example specifies a threshold of 128 and converts grayscale to a black and white mask using the specified value.

{button ,AL(`PP_Mask_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.MaskToMaskChannel (PHOTO-PAINT)

This command stores the active mask in an existing mask channel.

Syntax
.MaskToMaskChannel .MaskID=*long*

Syntax	Definition
.MaskID	Identifies the mask channel.

Example
```
.MaskToMaskChannel 0
```

The above example stores the active mask in the existing mask, "Mask5".

{button ,AL(`PP_Mask_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.MaskTranslate (PHOTO-PAINT)

This command translates/repositions the active mask.

Syntax

.MaskTranslate .ptX=*long*, .ptY=*long*

Syntax	Definition
.ptX	Specifies the X coordinate of the lower-left corner of the translated mask in pixels, relative to the origin.
.ptY	Specifies the Y coordinate of the lower-left corner of the translated mask in pixels, relative to the origin.

Example

```
.MaskTranslate 100, 100
```

The above example translates the mask to the new coordinates shown. The coordinates are expressed in pixels.

{button ,AL(`PP_Mask_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.MovieBackOne (PHOTO-PAINT)

This command backs up the movie one frame and displays it in the main image window.

Syntax
.MovieBackOne

Example
```
.MovieBackOne
```

The above example moves the movie back by one frame.

{button ,AL(`PP_Movie_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.MovieDeleteFrame (PHOTO-PAINT)

This command deletes the specified number of frames from a movie.

Syntax
.MovieDeleteFrame .FromFrame=*long*, .ToFrame=*long*

Syntax	Definition
.FromFrame	Specifies the first frame to be deleted from the movie.
.ToFrame	Specifies the last frame to be deleted from the movie.

Example

```
.MovieDeleteFrame 7, 9
```

The above example deletes movie frames number 7, 8, and 9.

{button ,AL(`PP_Movie_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.MovieForward (PHOTO-PAINT)

This command advances the movie to its final frame and displays it in the main image window.

Syntax
.MovieForward

Example
```
.MovieForward
```

The above example advances the movie to the last frame.

{button ,AL(`PP_Movie_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.MovieForwardOne (PHOTO-PAINT)

This command advances the movie one frame and displays it in the main image window.

Syntax
.MovieForwardOne

Example
```
.MovieForwardOne
```

The above example advances the movie by one frame.

{button ,AL(`PP_Movie_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.MovieGotoFrame (PHOTO-PAINT)

This command lets you go to a particular movie frame. The selected frame is immediately displayed in the main image window.

Syntax
.MovieGotoFrame .Frame=*long*

Syntax	Definition
.Frame	Specifies the frame number to which you want to advance.

Example

```
.MovieGotoFrame 5
```

The above example makes movie frame number 5 the active image.

```
.MovieGotoFrame 16
```

The above example makes movie frame number 16 the active image.

{button ,AL(`PP_Movie_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.MovieInsertFile (PHOTO-PAINT)

This command inserts any image file into an active/open movie file.

Syntax

.MovieInsertFile .FileName=*string*, .Left=*long*, .Top=*long*, .Right=*long*, .Bottom=*long*, .LoadType=*long*, .StartFrame=*long*, .Before=*boolean*

Syntax	Definition
.FileName	Specifies the name of the file to be inserted.
.Left	Specifies the X coordinate of the upper-left corner of the file to be inserted in pixels, relative to the origin.
.Top	Specifies the Y coordinate of the upper-left corner of the file to be inserted in pixels, relative to the origin.
.Right	Specifies the X coordinate of the lower-right corner of the file to be inserted in pixels, relative to the origin.
.Bottom	Specifies the Y coordinate of the lower-right corner of the file to be inserted in pixels, relative to the origin.
.LoadType	Specifies the image load type of the file: 0 = All 1 = Partial 2 = Resample 3 = Crop
.StartFrame	Specifies the insertion point of a frame or series of frames.
.Before	Set to TRUE (-1) to insert the selected frame(s) before the Start frame. Set to FALSE (0) to insert the selected frame(s) after the Start frame.

Example

```
.FileNew 640, 480, 1, 72, 72, 0, -1, 10, 0, 0, 0, 0, 255, 255, 255, 0
.MovieInsertFile "C:\COREL60\CUSTOM\TILES\TOWELF.PCX", 0, 0, 0, 0, 0, 1, 0
```

The above example inserts the file named "TOWELF.PCX", load type of all, after frame number 1.

{button ,AL(`PP_Movie_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.MovieInsertFrame (PHOTO-PAINT)

This command inserts one or more frames into a movie.

Syntax

.MovieInsertFrame .StartFrame=*long*, .NumberOfFrames=*long*, .Before=*boolean*, .CopyCurrent=*boolean*

Syntax	Definition
.StartFrame	Specifies the insertion point of a frame or series of frames.
.NumberOfFrames	Specifies the number of frames to be inserted at the designated insertion point.
.Before	Set to TRUE (-1) to insert the selected frame(s) before the Start frame. Set to FALSE (0) to insert the selected frame(s) after the Start frame.
.CopyCurrent	Set to TRUE (-1) to copy the current frame. Set to FALSE (0) to use the paper color .

Example

```
.MovieInsertFrame 5, 5, 0, 0
```

The above example inserts 5 movie frames after movie frane number 5, using the paper color.

```
.MovieInsertFrame 2, 8, -1, -1
```

The above example inserts 2 movie frames before movie frame number 8, copying the current frame.

{button ,AL(`PP_Movie_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.MovieMoveFrame (PHOTO-PAINT)

This command rearranges frames in a movie. You can move single or multiple frames to any point in the sequence of the movie.

Syntax

.MovieMoveFrame .FromFrame=*long*, .ToFrame=*long*, .MoveToFrame=*long*, .Before=*boolean*

Syntax	Definition
.FromFrame	Specifies the first frame included in the frame move.
.ToFrame	Specifies the last frame included in the frame move.
.MoveToFrame	Specifies the insertion point of the frame move. The frames selected for the move will be inserted before or after this frame depending on the status of the Before parameter.
.Before	Set to TRUE (-1) to move the selected frame(s) before the insertion point. Set to FALSE (0) to move the selected frame(s) after the insertion point.

Example

```
.MovieMoveFrame 5, 12, 1, 0
```

The above examples moves frames number 5 through 12 and repositions them after frame number 1.

```
.MovieMoveFrame 4, 6, 8, -1
```

The above example moves frames number 4, 5, and 6 and repositions them before frame number 8.

{button ,AL(`PP_Movie_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.MovieRewind (PHOTO-PAINT)

This command rewinds the movie to the first frame and displays it in the main image window.

Syntax
.MovieRewind

Example
```
.MovieRewind
```

The above example rewinds the movie back to the beginning.

{button ,AL(`PP_Movie_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.ObjectAlign (PHOTO-PAINT)

This command aligns multiple objects.

Syntax

.ObjectAlign .Horizontal=*long*, .Vertical=*long*, .Center=*boolean*, .Grid=*boolean*

Syntax	Definition
.Horizontal	Specifies the type of horizontal alignment. Horizontally aligns the selected objects to the center of the last object selected or to the grid: 0 = None 1 = Left 2 = Right 3 = Center
.Vertical	Specifies the type of vertical alignment. Vertically aligns the selected objects to the center of the last object selected or to the grid: 0 = None 1 = Top 2 = Bottom 3 = Center
.Center	Set to TRUE (-1) to align objects to the center of the page. Set to FALSE (0) to align objects to the grid.
.Grid	Set to TRUE (-1) to align objects to the grid. Set to FALSE (0) to align objects to the center of the page.

Note

You must choose at least one of the Horizontal or Vertical alignment options.

Example

```
.ObjectAlign 3, 3, -1, 0
```

The above example aligns the selected objects to the center of the page.

```
.ObjectAlign 2, 1, 0, -1
```

The above example aligns the selected objects vertically to the top and horizontally to the right, in reference to the grid.

{button ,AL(`PP_Object_Menu;;;;',0,"Defaultoverview",)} **Related Topics**

.ObjectClip (PHOTO-PAINT)

This command clips objects to the current mask.

Syntax
.ObjectClip

Example
```
.ObjectClip
```

The above example clips the active object.

{button ,AL(`PP_Object_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.ObjectCombine (PHOTO-PAINT)

This command combines two or more selected objects into a single object.

Syntax
.ObjectCombine

Example
```
.ObjectCombine
```

The above example combines the selected object(s) with the background.

{button ,AL(`PP_Object_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.ObjectCreate (PHOTO-PAINT)

This command creates an object using the shape and contents of the active mask; which effectively transforms a mask into an object.

Syntax

.ObjectCreate .PreserveImage=*boolean*

Syntax	Definition
.PreserveImage	Set to TRUE (-1) to copy the image inside the mask. Set to FALSE (0) to cut the image inside the mask.

Note

This command clears the original mask.

Example

```
.MaskRectangle 95, 46, 260, 225, 0, 0
.ObjectCreate 0
```

The above example creates an object from a mask.

{button ,AL(`PP_Object_Menu;;;;',0,"Defaultoverview",)} **Related Topics**

.ObjectDefringe (PHOTO-PAINT)

This command extends/spreads the colors in the interior to the edges.

Syntax
.ObjectDefringe .Amount=*long*

Syntax	Definition
.Amount	Specifies the width of the Defringe effect, expressed in pixels. Valid values range from 1 to 100.

Example

```
.ObjectDefringe 10
```

The above example applies a defringe to the object with a width of 10 pixels.

{button ,AL(`PP_Object_Menu;;;;',0,"Defaultoverview",)} **Related Topics**

.ObjectDelete (PHOTO-PAINT)

This command deletes the selected object(s).

Syntax
.ObjectDelete

Example
```
.ObjectDelete
```
The above example deletes the selected object(s).

{button ,AL(`PP_Object_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.ObjectDistort (PHOTO-PAINT)

This command distorts the shape of the selected object.

Syntax

.ObjectDistort .Corner1X=*long*, .Corner1Y=*long*, .Corner2X=*long*, .Corner2Y=*long*, .Corner3X=*long*, .Corner3Y=*long*, .Corner4X=*long*, .Corner4Y=*long*, .AntiAlias=*boolean*

Syntax	Definition
.Corner1X	Specifies the X coordinate of the new location of the upper-left corner of the object's original bounding rectangle in pixels, relative to the origin.
.Corner1Y	Specifies the Y coordinate of the new location of the upper-left corner of the object's original bounding rectangle in pixels, relative to the origin.
.Corner2X	Specifies the X coordinate of the new location of the upper-right corner of the object's original bounding rectangle in pixels, relative to the origin.
.Corner2Y	Specifies the Y coordinate of the new location of the upper-right corner of the object's original bounding rectangle in pixels, relative to the origin.
.Corner3X	Specifies the X coordinate of the new location of the lower-right corner of the object's original bounding rectangle in pixels, relative to the origin.
.Corner3Y	Specifies the Y coordinate of the new location of the lower-right corner of the object's original bounding rectangle in pixels, relative to the origin.
.Corner4X	Specifies the X coordinate of the new location of the lower-left corner of the object's original bounding rectangle in pixels, relative to the origin.
.Corner4Y	Specifies the Y coordinate of the new location of the lower-left corner of the object's original bounding rectangle in pixels, relative to the origin.
.AntiAlias	Set to TRUE (-1) to apply anti-aliasing. Set to FALSE (0) to disable anti-aliasing.

Example

```
.ObjectDistort 47, 180, 187, 141, 203, 15, 47, 15, 0
```

The above example distorts the selected object by the specified parameters.

{button ,AL(`PP_Object_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.ObjectDuplicate (PHOTO-PAINT)

This command duplicates the selected object(s). The new objects are placed in front of existing objects and remain selected. The original objects are deselected.

Syntax
.ObjectDuplicate

Example
```
.ObjectDuplicate
```

The above example creates a duplicate of the selected object.

{button ,AL(`PP_Object_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.ObjectFeather (PHOTO-PAINT)

This command feathers the edges of the selected object(s). Feathering is the blending of the edge of an object with an underlying image.

Syntax

.ObjectFeather .Width=*long*, .Type=*long*

Syntax	Definition
.Width	Specifies the number of pixels that are used in the feathering transition. Valid width values range from 1 to 200.
.Type	Specifies the intensity of the feathered edge. A soft edge produces a more gradual blending between the object and the background than a hard edge. 0 = Hard edge 1 = Soft edge

Example

```
.ObjectFeather 10, 0
```

The above example sets the feathering width to 10 pixels, applying a hard edge.

{button ,AL(`PP_Object_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.ObjectFlipHorizontal (PHOTO-PAINT)

This command flips the selected object(s) horizontally.

Syntax
.ObjectFlipHorizontal

Example
```
.ObjectFlipHorizontal
```

The above example flips the selected object horizontally.

{button ,AL(`PP_Object_Menu;;;;',0,"Defaultoverview",)} Related Topics

.ObjectFlipVertical (PHOTO-PAINT)

This command flips the selected object(s) vertically.

Syntax
.ObjectFlipVertical

Example
```
.ObjectFlipVertical
```

The above example flips the selected object vertically.

{button ,AL(`PP_Object_Menu;;;;',0,"Defaultoverview",)} Related Topics

.ObjectGroup (PHOTO-PAINT)

This command groups all selected objects together so that they can be selected and manipulated as a single object.

Syntax
.ObjectGroup

Example
```
.ObjectGroup
```

The above example collects/groups selected objects together.

{button ,AL(`PP_Object_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.ObjectMerge (PHOTO-PAINT)

This command merges the selected object(s) or all objects with the image. The object(s) become(s) part of the image and cannot be selected again.

Syntax

.ObjectMerge .All=*boolean*

Syntax	Definition
.All	Set to TRUE (-1) to merge all objects. Set to FALSE (0) to merge selected objects only.

Example

```
.ObjectMerge -1
```

The above example merges all objects.

{button ,AL(`PP_Object_Menu;;;;',0,"Defaultoverview",)} **Related Topics**

.ObjectMergeMode (PHOTO-PAINT)

This command sets the merge mode of the selected object(s).

Syntax
.ObjectMergeMode .MergeMode=*long*

Syntax	Definition
.MergeMode	Specifies the Merge Mode: 0 = Normal 1 = Add 2 = Subtract 3 = Difference 4 = Multiply 5 = Divide 6 = Lighter 7 = Darker 8 = Texturize 9 = Color 10 = Hue 11 = Saturation 12 = Lum 13 = Invert 14 = And 15 = Or 16 = Xor 17 = Red 18 = Green 19 = Blue 20 = Cyan 21 = Magenta 22 = Yellow 23 = Black

Note

The modes which may be used depend on the image type.

Example

```
.ObjectMergeMode 11
```

The above example sets the object merge mode to Saturation.

{button ,AL(`PP_Object_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.ObjectName (PHOTO-PAINT)

This command sets the name of the specified object.

Syntax
.ObjectName .ObjectID=*long*, .Name=*string*

Syntax	Definition
.ObjectID	Identifies the object.
.Name	Specifies the new name of the object.

Example

```
.ObjectName 1, "Object1"
```

The above example names object 1 "Dave".

{button ,AL(`PP_Object_Menu;;;;',0,"Defaultoverview",)} **Related Topics**

.ObjectOpacity (PHOTO-PAINT)

This command sets the opacity of the selected object(s).

Syntax
.ObjectOpacity .Amount=*long*

Syntax	Definition
.Amount	Specifies the opacity. Valid values range from 0 to 99 percent.

Example
```
.ObjectOpacity 50
```
The above example sets the object's opacity to 50%.

{button ,AL(`PP_Object_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.ObjectOrder (PHOTO-PAINT)

This command controls the stacking order of selected object(s) in the image.

Syntax
.ObjectOrder .Order=*long*

Syntax	Definition
.Order	Specifies the stacking order of selected object(s): 0 = To front 1 = To back 2 = Forward 3 = Back 4 = Reverse

Note

If multiple objects are selected, they retain their relative order (except in the case of reverse ordering which reverses the order of the selected objects).

For reverse ordering, at least two objects must be selected.

Example

```
.ObjectOrder 1
```

The above example orders the selected object(s) to the front of the document.

```
.ObjectOrder 3
```

The above example orders the selected object(s) back one.

{button ,AL(`PP_Object_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.ObjectRemoveMatte (PHOTO-PAINT)

This command removes the black or white matte (border pixels) sometimes found along the edges of an object when an object is cut away from a black or white background.

Syntax

.ObjectRemoveMatte .White=*boolean*

Syntax	Definition
.White	Set to TRUE (-1) to remove white matting. Set to FALSE (0) to remove black matting.

Example

```
.ObjectRemoveMatte -1
```

The above example removes the black or white matting from an object.

{button ,AL(`PP_Object_Menu;;;;',0,"Defaultoverview",)} **Related Topics**

.ObjectRotate (PHOTO-PAINT)

This command rotates the selected object(s) by the specified angle, about the given rotation point.

Syntax

.ObjectRotate .XCenter=*long*, .YCenter=*long*, .Angle=*long*, .AntiAlias=*boolean*

Syntax	Definition
.XCenter	Specifies the X coordinate of the center of rotation in pixels, relative to the origin.
.YCenter	Specifies the Y coordinate of the center of rotation in pixels, relative to the origin.
.Angle	Specifies the angle of rotation in tenths of degrees, relative to the center of the object. Positive numbers result in counter-clockwise rotation, negative numbers result in clockwise rotation. e.g., 45 degrees clockwise = -450
.AntiAlias	Set to TRUE (-1) to apply anti-aliasing. Set to FALSE (0) to disable anti-aliasing.

Note

You can use the ANGLECONVERT function to specify angle measurements.

Example

```
.ObjectRotate 180, 252, -450, -1
```

The above example rotates the selected object 45 degrees clockwise relative to the object's center, applying anti-aliasing.

{button ,AL(`PP_Object_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.ObjectSelect (PHOTO-PAINT)

This command selects or deselects the specified object.

Syntax
.ObjectSelect .ObjectID=*long*, .Selected=*boolean*

Syntax	Definition
.ObjectID	Identifies the selected object.
.Selected	Set to TRUE (-1) to select the object. Set to FALSE (0) to deselect the object.

Note

Selecting or deselecting an object in a group selects/deselects all objects in the group.

Example

```
.ObjectSelect 2, -1
```

The above example selects object number 2.

{button ,AL(`PP_Object_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.ObjectSelectAll (PHOTO-PAINT)

This command selects all objects.

Syntax
.ObjectSelectAll

Example
```
.ObjectSelectAll
```
The above example selects all objects.

{button ,AL(`PP_Object_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.ObjectSelectNone (PHOTO-PAINT)

This command deselects any selected objects.

Syntax
.ObjectSelectNone

Example
```
.ObjectSelectNone
```

The above example deselects all selected objects.

{button ,AL(`PP_Object_Menu;;;;',0,"Defaultoverview",)} Related Topics

.ObjectSkew (PHOTO-PAINT)

This command skews the selected object(s).

Syntax

.ObjectSkew .ptX=*long*, .ptY=*long*, .HorzAngle=*long*, .VertAngle=*long*, .AntiAlias=*boolean*

Syntax	Definition
.ptX	Specifies the X coordinate of the lower-left corner of the skewed object.
.ptY	Specifies the Y coordinate of the lower-left corner of the skewed object.
.HorzAngle	Specifies the horizontal skew angle in tenths of degrees. Positive numbers result in counter-clockwise skew, negative numbers result in clockwise skew.
.VertAngle	Specifies the vertical skew angle in tenths of degrees. Positive numbers result in counter-clockwise skew, negative numbers result in clockwise skew.
.AntiAlias	Set to TRUE (-1) to apply anti-aliasing. Set to FALSE (0) to disable anti-aliasing.

Note

You can use the ANGLECONVERT function to specify angle measurements.

Example

```
.ObjectSkew -300, -300, -125, -53, 0
```

The above example skews the selected object by 30 degrees horizontally and 30 degrees vertically.

{button ,AL(`PP_Object_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.ObjectStretch (PHOTO-PAINT)

This command stretches the selected object(s).

Syntax

.ObjectStretch .Left=*long*, .Top=*long*, .Right=*long*, .Bottom=*long*, .AntiAlias=*boolean*

Syntax	Definition
.Left	Specifies the X coordinate of the new upper-left corner of the object's bounding rectangle in pixels, relative to the origin.
.Top	Specifies the Y coordinate of the new upper-left corner of the object's bounding rectangle in pixels, relative to the origin.
.Right	Specifies the X coordinate of the new lower-right corner of the object's bounding rectangle in pixels, relative to the origin.
.Bottom	Specifies the Y coordinate of the new lower-right corner of the object's bounding rectangle in pixels, relative to the origin.
.AntiAlias	Set to TRUE (-1) to apply anti-aliasing. Set to FALSE (0) to disable anti-aliasing.

Example

```
.ObjectStretch 51, 461, 282, 151, -1
```

The above example stretches the bottom edge of the object to the new specified coordinate, applying anti-aliasing.

{button ,AL(`PP_Object_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.ObjectTranslate (PHOTO-PAINT)

This command translates/repositions selected object(s).

Syntax

.ObjectTranslate .ptX=*long*, .ptY=*long*, .Copy=*boolean*

Syntax	Definition
.ptX	Specifies the X coordinate of the lower-left corner of the object's bounding rectangle in pixels, relative to the origin.
.ptY	Specifies the Y coordinate of the lower-left corner of the object's bounding rectangle in pixels, relative to the origin.
.Copy	Set to TRUE (-1) to make a copy of the object. Set to FALSE (0) to translate the object.

Example

```
.ObjectTranslate 0, 0, 0
```

The above example moves the object to the specified new coordinates of the lower-left corner of the object's bounding rectangle.

```
.ObjectTranslate -52, 0, 0
```

The above example moves the object to the specified new coordinates of the lower-left corner of the object's bounding rectangle.

{button ,AL(`PP_Object_Menu;;;;',0,"Defaultoverview",)} **Related Topics**

.ObjectUngroup (PHOTO-PAINT)

This command breaks up the selected group of objects into its component objects.

Syntax
.ObjectUngroup

Example
```
.ObjectUngroup
```
The above example reverses the ObjectGroup command, and frees/ungroups the grouped objects.

{button ,AL(`PP_Object_Menu;;;;',0,"Defaultoverview",)} Related Topics

.ObjectVisible (PHOTO-PAINT)

This command hides or shows the specified object.

Syntax
.ObjectVisible .ObjectID=*long*, .Visible=*boolean*

Syntax	Definition
.ObjectID	Identifies the specified object.
.Visible	Set to TRUE (-1) to make object visible. Set to FALSE (0) to hide object.

Note

Hiding an object deselects it (if it was selected) and makes it uneditable.

Example

```
.ObjectVisible 2, -1
```

The above example makes the specified object visible.

{button ,AL(`PP_Object_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.PenSettings (PHOTO-PAINT)

This command specifies settings for various drawing tools.

Syntax

.PenSettings .ToolID=*long*, .Width=*long*, .Flatten=*long*, .Rotate=*long*, .NibShape=*long*, .Transparency=*long*, .MergeMode=*long*

Syntax	Definition
.ToolID	The ToolID specifies the type of tool to activate: 0 = Pick Tool 1 = Rectangle Mask 2 = Ellipse Masks 3 = Freehand Mask 4 = Mask Noose 5 = Mask Wand 6 = Mask Brush 7 = Node 8 = Mask Pick 9 = Crop 12 = Hand 13 = Eyedropper 14 = LC Undo 15 = Eraser 16 = Clear Eraser 17 = Line 18 = Curve 19 = Pen 20 = Round Rectangle 21 = Oval 22 = Polygon 23 = Text 24 = Fill 25 = Brush 26 = Effect 27 = Clone
.Width	Specifies the size of the pen in pixels.
.Flatten	Specifies the flatness of the brush. Valid values range from 0 to 99 percent.
.Rotate	Specifies the amount of rotation applied to a round or square brush, in degrees.
.NibShape	Specifies the shape of the pen nib: 0 = Circle 1 = Square 2 = Custom
.Transparency	Specifies the amount of transparency to apply.
.MergeMode	Specifies the Merge Mode: 0 = Normal 1 = Add 2 = Subtract 3 = Difference 4 = Multiply 5 = Divide 6 = Lighter 7 = Darker 8 = Texturize 9 = Color 10 = Hue 11 = Saturation 12 = Lum 13 = Invert 14 = And 15 = Or 16 = Xor 17 = Red

18 = Green
19 = Blue
20 = Cyan
21 = Magenta
22 = Yellow
23 = Black

Note

Available merge modes depend on the image type.

Example

```
.PenSettings 20, 20, 100, 0, 0, 0, 0
```

The above example sets the pen tool to rectangles with a width of 20 pixels.

{button ,AL(`PP_Tools_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.Rectangle (PHOTO-PAINT)

This command draws hollow/filled rectangles and rounded rectangles.

Syntax

.Rectangle .Left=*long*, .Top=*long*, .Right=*long*, .Bottom=*long*

Syntax	Definition
.Left	Specifies the logical X coordinate of the upper-left corner of the rectangle in pixels, relative to the origin.
.Top	Specifies the Y coordinate of the upper-left corner of the rectangle in pixels, relative to the origin.
.Right	Specifies the X coordinate of the lower-right corner of the rectangle in pixels, relative to the origin.
.Bottom	Specifies the Y coordinate of the lower-right corner of the rectangle in pixels, relative to the origin.

Note

The .Rectangle command must be preceded by the appropriate tool settings commands:
.SetPaintColor
.PenSettings
.ShapeSettings.

Example

```
.Rectangle 41, 154, 173, 75
```

The above example creates a rectangle with the upper-left corner at the point (41, 154) and the lower-right corner at the point (173, 75). These coordinates are expressed in pixels.

```
.Rectangle 94, 84, 275, 229
```

The above example creates a rectangle with the upper-left corner at the point (94, 84) and the lower-right corner at the point (275, 229). These coordinates are expressed in pixels.

{button ,AL(`PP_Draw_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.RenderText (PHOTO-PAINT)

This command lets you add text to your picture.

Syntax
.RenderText .ptX=*long*, .ptY=*long*, .Text=*string*

Syntax	Definition
.ptX	Specifies the X coordinate of the point to insert text in pixels, relative to the origin.
.ptY	Specifies the Y coordinate of the point to insert text in pixels, relative to the origin.
.Text	Specifies the text to be inserted.

Note

The .RenderText command must be preceded by the appropriate tool settings commands:
.SetPaintColor
.TextSettings.

Example

```
.RenderText 10, 290, "Hello World"
```

The above example inserts the text string "Hello World" at the point (10, 290). The coordinates are expressed in pixels.

{button ,AL(`PP_Draw_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.ImageSetChannel (PHOTO-PAINT)

This command selects the image channel or channels to be edited. Valid only for RGB and CMYK images. A channel is similar to a plate in the printing process.

Syntax

.ImageSetChannel .Channel=*long*

Syntax	Definition
.Channel	Specifies the channel(s) to be set. For RGB images: 0 = Blue 1 = Green 2 = Red -1 = All channels For CMYK images: 0 = Cyan 1 = Magenta 2 = Yellow 3 = Black -1 = All channels

Example

```
.ImageSetChannel 2
```

The above example sets/selects the red image channel to be edited.

{button ,AL(`PP_Image_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.SetPaintColor (PHOTO-PAINT)

This command lets you set the paint color used by the Drawing Tool (e.g., Pen, Pencil).

Syntax

.SetPaintColor .ColorModel=*long*, .Color1=*long*, .Color2=*long*, .Color3=*long*, .Color4=*long*

Syntax	Definition
.ColorModel	Specifies the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Specifies the first color component for .ColorModel. For example, Red is the first color component for RGB. Click + for valid value ranges.

.Color2 Specifies the second color component for .ColorModel. For example, Green is the second color component for RGB. Click + for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.

.Color3	Specifies the third color component for .ColorModel. For example, Blue is the third color component for RGB. Click + for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.

.Color4 Specifies the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. Click + for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.

Example

```
.SetPaintColor 5, 102, 102, 255, 0
```

The above example uses the RGB color mode and sets the drawing color to blue.

{button ,AL(`PP_Tools_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.SetPaperColor (PHOTO-PAINT)

This command sets the paper color used when clearing or erasing the image.

Syntax

.SetPaperColor .ColorModel=*long*, .Color1=*long*, .Color2=*long*, .Color3=*long*, .Color4=*long*

Syntax	Definition
.ColorModel	Specifies the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Specifies the first color component for .ColorModel. For example, Red is the first color component for RGB. Click + for valid value ranges.

.Color2 Specifies the second color component for .ColorModel. For example, Green is the second color component for RGB. Click + for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.

.Color3	Specifies the third color component for .ColorModel. For example, Blue is the third color component for RGB. Click + for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.

.Color4 Specifies the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. Click + for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.

Example

```
.SetPaperColor 5, 0, 0, 255, 0
```

The above example sets the paper color to blue.

{button ,AL(`PP_Tools_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.ShapeSettings (PHOTO-PAINT)

This command specifies the settings for drawing rectangles, circles, polygons, and line tools.

Syntax

.ShapeSettings .Roundness=*long*, .Joints=*long*, .Width=*long*, .Fill=*boolean*, .AntiAlias=*boolean*

Syntax	Definition
.Roundness	Specifies the roundness of the lines.
.Joints	Specifies the shape of the line joints: 0 = Butt 1 = Filled 2 = Round 3 = Point
.Width	Specifies the width of the line or border in pixels.
.Fill	Set to TRUE (-1) to enable the Fill option. Set to FALSE (0) to disable the Fill option.
.AntiAlias	Set to TRUE (-1) to apply anti-aliasing. Set to FALSE (0) to disable anti-aliasing.

Example

```
.ShapeSettings 0, 0, 20, -1, 0
```

The above example sets the line tool to a width of 20 pixels with the fill option enabled.

{button ,AL(`PP_Tools_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.SnapToGrid (PHOTO-PAINT)

This command turns Snap to Grid on and off. Working with the grid on makes it easier to align and position objects accurately. It also allows you to size objects precisely. The grid is not printed.

Syntax

.SnapToGrid .SnapOn=*boolean*

Syntax	Definition
.SnapOn	Set to TRUE (-1) to enable Snap to Grid. Set to FALSE (0) to disable Snap to Grid.

Example

```
.SnapToGrid -1
```

The above example enables the snap to grid option and aligns selected objects to the grid.

{button ,AL(`PP_View_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.StartDraw (PHOTO-PAINT)

This command begins a freehand tool command such as drawing, masking, erasing, and filling.

Syntax

.StartDraw .ptX=*long*, .ptY=*long*

Syntax	Definition
.ptX	Specifies the X coordinate of the point to start drawing in pixels, relative to the origin.
.ptY	Specifies the Y coordinate of the point to start drawing in pixels, relative to the origin.

Note

The .StartDraw command must be followed by one or more .ContinueDraw commands, and one .EndDraw command.

The .StartDraw command must be preceded by the appropriate tool settings commands.

Example

```
.SetPaintColor 5, 153, 0, 204, 0
.PenSettings 17, 5, 0, 0, 0, 0, 0
.ShapeSettings 0, 0, 5, -1, -1
.StartDraw 162, 365
.ContinueDraw 308, 203
.ContinueDraw 76, 203
.ContinueDraw 162, 365
.EndDraw
```

The above example draws a triangle with points at coordinates (162, 365), (308, 203), and (76, 203). These coordinates are expressed in pixels.

```
.StartDraw 140, 169
.ContinueDraw 138, 168
.ContinueDraw 136, 166
.ContinueDraw 130, 160
...
...
...
.ContinueDraw 136, 167
.ContinueDraw 137, 167
.EndDraw
```

The above example creates a Lasso Mask.

{button ,AL(`PP_Draw_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.TextSettings (PHOTO-PAINT)

This command specifies the settings for text to be drawn with the RenderText command.

Syntax

.TextSettings .Bold=*boolean*, .Italic=*boolean*, .Underline=*boolean*, .Alignment=*long*, .FontIndex=*long*, .FontSize=*long*, .AntiAlias=*boolean*

Syntax	Definition
.Bold	Set to TRUE (-1) to change text to bold.
.Italic	Set to TRUE (-1) to change text to italics.
.Underline	Set to TRUE (-1) to underline text.
.Alignment	Specifies the text alignment: 0 = Left 1 = Center 2 = Right
.FontIndex	Specifies the font index. The font index will vary depending on your system. Valid values start at 0.
.FontSize	Specifies the font size, in points.
.AntiAlias	Set to TRUE (-1) to apply anti-aliasing. Set to FALSE (0) to disable anti-aliasing.

Example

```
.TextSettings -1, -1, 0, 1, 2, 12, -1
```

The above example sets the font to bold, italic, 12 point, centered, "Arial", applying anti-aliasing.

{button ,AL(`PP_Tools_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.ToleranceSettings (PHOTO-PAINT)

This command specifies the range of colors to be selected using the Magic Wand and Fill tools.

Syntax

.ToleranceSettings .ToleranceMode=*long*, .Normal=*long*, .Hue=*long*, .Saturation=*long*, .Brightness=*long*

Syntax	Definition
.ToleranceMode	Color Tolerance is an adjustment for controlling how many shades of a selected color are masked. Selecting a higher tolerance value causes a wider range of colors to be masked; a lower tolerance value causes a smaller range of colors to be masked. 0 = Normal 1 = HSB
.Normal	Set to 0 to enable normal mode. Set to 1 to enable HSB mode. Note: If Normal mode is selected, set Hue, Saturation, and Brightness to 0.
.Hue	Specifies the hue. In the HSB color model, hue is the main attribute in a color that distinguishes it from other colors. Blue, green and red, for example, are all hues.
.Saturation	Specifies the saturation. Saturation is the purity of a color. The HSB color model uses Saturation as a component that determines the purity or intensity of a color. The more colors used to mix a color, the duller the color looks.
.Brightness	Specifies the brightness. In the HSB color model, the component that determines the amount of black in a color.

Example

```
.ToleranceSettings 1, 10, 7, 8, 9
```

The above example sets the mode to HSB, the Hue to 7, Saturation to 8, and Brightness to 9.

{button ,AL(`PP_Tools_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.ToolSettings (PHOTO-PAINT)

This command selects an existing paint brush, effect, or clone tool.

Syntax

.ToolSettings .ToolID=*long*, .BrushID=*long*, .TypeID=*long*

Syntax	Definition
.ToolID	Specifies the type of tool. 25 = Paint 26 = Effect 27 = Clone
.BrushID	Specifies the brush. If ToolID = 25, BrushID and TypeID can be specified, respectively, as: 0 = Art Brush, 0 = Custom Art Brush 0 = Art Brush, 1 = Thick 0 = Art Brush, 2 = Thin 0 = Art Brush, 3 = Flat 0 = Art Brush, 4 = Quick Doodler 0 = Art Brush, 5 = Hard edges 1 = Airbrush, 0 = Custom Airbrush 1 = Airbrush, 1 = Graffiti 2 = Spraycan, 0 = Custom Spraycan 2 = Spraycan, 1 = Power Sprayer 3 = Pencil, 0 = Custom 3 = Pencil, 1 = 2B 3 = Pencil, 2 = HB 4 = Ball Point, 0 = Custom Ball Point 4 = Ball Point, 1 = Ball Point 5 = Calligraphy, 0 = Custom Calligraphy 5 = Calligraphy, 0 = Calligraphy 6 = Fine Felt Tip, 0 = Custom Felt Tip 6 = Fine Felt Tip, 0 = Fine Felt Tip 7 = Marker, 0 = Custom Marker 7 = Marker, 0 = Felt Marker 8 = Hi-Liter, 0 = Custom Hi-Liter 8 = Hi-Liter, 0 = Hi-Liter 9 = Chalk, 0 = Custom Chalk 9 = Chalk, 0 = Chalk 10 = Crayon, 0 = Custom Crayon 10 = Crayon, 0 = Crayon 11 = Charcoal, 0 = Custom Charcoal 11 = Charcoal, 0 = Charcoal 12 = Pastel, 0 = Custom Pastel 12 = Pastel, 0 = Pastel 13 = Watery, 0 = Custom Watery 13 = Watery, 0 = Oil 13 = Watery, 0 = Watercolor 13 = Watery, 0 = Water Smudge 14 = Artistic, 0 = Custom Artistic 14 = Artistic, 0 = Pointillism 14 = Artistic, 0 = Impressionism If ToolID = 26, BrushID and TypeID can be specified, respectively, as : 0 = Smear, 0 = Custom Smear 0 = Smear, 0 = Pointy Smear 0 = Smear, 0 = Thick Smear 0 = Smear, 0 = Motion Blur 1 = Smudge, 0 = Custom Smudge 1 = Smudge, 0 = Smudge A Little 1 = Smudge, 0 = Smudge A Lot 2 = Brightness, 0 = Custom Brightness 2 = Brightness, 0 = Brighten 2 = Brightness, 0 = Darken 3 = Contrast, 0 = Custom Contrast 3 = Contrast, 0 = Decrease Contrast

3 = Contrast, 0 = Increase Contrast
4 = Hue, 0 = Custom Hue
4 = Hue, 0 = Hue
5 = Hue Replacer, 0 = Custom Hue Replacer
5 = Hue Replacer, 0 = Hue Replacer
6 = Sponge, 0 = Custom Sponge
6 = Sponge, 0 = Sponge Add
6 = Sponge, 0 = Sponge Remove
7 = Tint, 0 = Custom Tint
7 = Tint, 0 = Tint Lightly
7 = Tint, 0 = Tint A Lot
8 = Blend, 0 = Custom Blend
8 = Blend, 0 = Blend A Little
8 = Blend, 0 = Blend A Lot
9 = Sharpen, 0 = Custom Sharpen
9 = Sharpen, 0 = Sharpen A Little
9 = Sharpen, 0 = Sharpen A Lot

If ToolID = 27, BrushID and TypeID can be specified, respectively, as :
0 = Normal, 0 = Custom Normal
0 = Normal, 0 = Normal Clone
1 = Vango, 0 = Custom Impressionism
1 = Vango, 0 = Impressionism Clone
2 = Seurat, 0 = Custom Pointillism
2 = Seurat, 0 = Pointillism Clone
3 = From Saved, 0 = Custom From Saved
3 = From Saved, 0 = Light Eraser
3 = From Saved, 0 = Eraser
3 = From Saved, 0 = Scrambler

.TypeID Specifies the brush type, which varies depending on the tool type and the brush selected. See the BrushID settings above.

Note

To ensure that script behavior doesn't change, the brush settings shoild be fully specified using .BrushSettings, .NibSettings, .BrushNibSettings, and .BrushTextureSettings.

Example

```
.ToolSettings 25, 0, 4
```

The above example sets the Brush tool to Art Brush, Quick Doodler type.

{button ,AL(`PP_Tools_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.SetVisible (PHOTO-PAINT)

This command controls whether the PHOTO-PAINT application is visible or hidden.

Syntax
.SetVisible .Show=*boolean*

Syntax	Definition
.Show	Set to TRUE (-1) to show the Corel PHOTO-PAINT application. Set to FALSE (0) to hide the application.

Example

```
.SetVisible -1
```

The above example shows the Corel PHOTO-PAINT application.

{button ,AL(`PP_View_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.NibSettings (PHOTO-PAINT)

This command is used to select custom nibs.

Syntax
.NibSettings .FileName=*string*, .NibIndex=*long*

Syntax	Definition
.FileName	Specifies the name of the Nib file.
.NibIndex	Identifies the Nib to load. Refer to the Nibs roll-up for more details.

Example

```
.NibSettings "C:\COREL60\PHOTOPNT\PNTBRUSH.MSK", -1
```

The above example selects a custom nib from the specified file.

{button ,AL(`PP_Tools_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.GetChannelCount (PHOTO-PAINT)

This function returns a value indicating the number of Mask Channels saved.

Syntax
.GetChannelCount

Example

```
.MaskChannelAdd "Mask 1"
.MaskChannelAdd "Mask 2"
cnt& = .GetChannelCount()
MESSAGE cnt&
```

The above example gets the number of mask channels. For this example the variable cnt equals 2.

{button ,AL(`PP_Get_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.GetChannelName (PHOTO-PAINT)

This function returns a string containing the name of the specified channel.

Syntax
.GetChannelName .ChannelID=*long*

Syntax	Definition
.ChannelID	Specifies the channel.

Example

```
.MaskChannelName 1, "Mask 5"
cname$ = .GetChannelName (1)
MESSAGE cname$
```

The above example returns the name of the first channel, which was named "Channel 5".

{button ,AL(`PP_Get_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.GetPaintVersion (PHOTO-PAINT)

This function returns a string containing the version information.

Syntax
.GetPaintVersion

Example

```
ver$ = .GetPaintVersion()
MESSAGE ver$
```

The above example gets the current version information.

{button ,AL(`PP_Get_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.BrushTextureSettings (PHOTO-PAINT)

This command sets the attributes for the brush texture.

Syntax

.BrushTextureSettings .TextureFile=*string*, .BrushTexture=*long*, .EdgeTexture=*long*, .Bleed=*long*, .SustainColor=*long*, .Smoothing=*long*, .AntiAlias=*boolean*

Syntax	Definition
.TextureFile	Specifies the name of the Texture file.
.BrushTexture	Specifies the texture of the brush, which controls the grainy surface quality produced by a brush. Increase the value to make the brush stroke more coarse. Valid values range from 0 to 100.
.EdgeTexture	Specifies the texture of the brush edge, which controls the amount of texturing along the edges of a brush. Valid values range from 0 to 100.
.Bleed	Specifies the amount of bleed, which controls the amount of color diffusion when two or more colors are combined (as when red paint, for example, contacts blue). Bleed is especially useful when working with watercolors where interesting effects may be achieved by blending two or more colors together. Valid values range from 0 to 100.
.SustainColor	The Sustain Color control works in tandem with the Bleed control. The Bleed control works in tandem with the Transparency control. Consequently, you must enter a Bleed and Transparency when entering a Sustain Color value. Sustain Color retains brush paint color when painting over a colored background while applying bleed to the brush. Typically, using Bleed, the brush will eventually (during the course of an extended brush stroke) run out of paint and simple smear the background/image color with the brush. With Sustain Color, traces of the paint color remain throughout the brush stroke. Valid values range from 0 to 100.
.Smoothing	Specifies the amount of brush stroke smoothing, in pixels. Smoothing helps to create a more flowing and fluid paint stroke by smoothing out the jagged and/or sharp angles while you paint. Valid values range from 0 to 25.
.AntiAlias	Set to TRUE (-1) to apply anti-aliasing. Set to FALSE (0) to disable anti-aliasing.

Example

```
.BrushTextureSettings , 0, 0, 0, 0, 10, -1
```

The above example sets the brush texture to the default texture with 10 percent smoothing and anti-aliasing enabled.

{button ,AL(`PP_Tools_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.BrushDabSettings (PHOTO-PAINT)

This command sets the attributes for brush dabs.

Syntax

.BrushDabSettings .Dabs=*long*, .Spacing=*long*, .Spread=*long*, .FadeOut=*long*, .Hue=*long*, .Saturation=*long*, .Luminance=*long*

Syntax	Definition
.Dabs	Specifies the number of dabs in the brush stroke. Valid values range from 1 to 25.
.Spacing	Specifies the spacing between dabs, in pixels. A spacing of zero creates a solid line.
.Spread	Specifies the distance between brush strokes in pixels. Higher values make the distance between the strokes greater.
.FadeOut	Specifies the length of the brush stroke before it fades out entirely. A value of zero sets the fade out to none. Valid values range from 1 to 100.
.Hue	Specifies the hue (a particular color) in pixels. Valid values range from 0 to 100.
.Saturation	Specifies the saturation (amount of a color). Valid values range from 0 to 100.
.Luminance	Specifies the brightness of a color. Valid values range from 0 to 100.

Example

```
.BrushDabSettings 4, 20, 90, 50, 40, 60, 50
```

The above example sets the brush to 4 dabs per stroke, with a spacing of 20, a spread of 90, fade out of 50, hue 40, saturation 50, luminance 60.

{button ,AL(`PP_Tools_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.BrushSettings (PHOTO-PAINT)

This command sets the attributes for the brush.

Syntax

.BrushSettings .ToolID=*long*, .BrushID=*long*, .TypeID=*long*, .MergeMode=*long*, .Amount=*long*, .NibShape=*long*, .Size=*long*, .Transparency=*long*, .Rotate=*long*, .Flatten=*long*, .SoftEdge=*long*

Syntax	Definition
.ToolID	Specifies the type of tool. 25 = Paint 26 = Effect 27 = Clone
.BrushID	Specifies the brush. If ToolID = 25, BrushID and TypeID can be specified, respectively, as: 0 = Art Brush, 0 = Custom Art Brush 0 = Art Brush, 1 = Thick 0 = Art Brush, 2 = Thin 0 = Art Brush, 3 = Flat 0 = Art Brush, 4 = Quick Doodler 0 = Art Brush, 5 = Hard edges 1 = Airbrush, 0 = Custom Airbrush 1 = Airbrush, 1 = Graffiti 2 = Spraycan, 0 = Custom Spraycan 2 = Spraycan, 1 = Power Sprayer 3 = Pencil, 0 = Custom 3 = Pencil, 1 = 2B 3 = Pencil, 2 = HB 4 = Ball Point, 0 = Custom Ball Point 4 = Ball Point, 1 = Ball Point 5 = Calligraphy, 0 = Custom Calligraphy 5 = Calligraphy, 0 = Calligraphy 6 = Fine Felt Tip, 0 = Custom Felt Tip 6 = Fine Felt Tip, 0 = Fine Felt Tip 7 = Marker, 0 = Custom Marker 7 = Marker, 0 = Felt Marker 8 = Hi-Liter, 0 = Custom Hi-Liter 8 = Hi-Liter, 0 = Hi-Liter 9 = Chalk, 0 = Custom Chalk 9 = Chalk, 0 = Chalk 10 = Crayon, 0 = Custom Crayon 10 = Crayon, 0 = Crayon 11 = Charcoal, 0 = Custom Charcoal 11 = Charcoal, 0 = Charcoal 12 = Pastel, 0 = Custom Pastel 12 = Pastel, 0 = Pastel 13 = Watery, 0 = Custom Watery 13 = Watery, 0 = Oil 13 = Watery, 0 = Watercolor 13 = Watery, 0 = Water Smudge 14 = Artistic, 0 = Custom Artistic 14 = Artistic, 0 = Pointillism 14 = Artistic, 0 = Impressionism If ToolID = 26, BrushID and TypeID can be specified, respectively, as : 0 = Smear, 0 = Custom Smear 0 = Smear, 0 = Pointy Smear 0 = Smear, 0 = Thick Smear 0 = Smear, 0 = Motion Blur 1 = Smudge, 0 = Custom Smudge 1 = Smudge, 0 = Smudge A Little 1 = Smudge, 0 = Smudge A Lot 2 = Brightness, 0 = Custom Brightness 2 = Brightness, 0 = Brighten 2 = Brightness, 0 = Darken 3 = Contrast, 0 = Custom Contrast

3 = Contrast, 0 = Decrease Contrast
3 = Contrast, 0 = Increase Contrast
4 = Hue, 0 = Custom Hue
4 = Hue, 0 = Hue
5 = Hue Replacer, 0 = Custom Hue Replacer
5 = Hue Replacer, 0 = Hue Replacer
6 = Sponge, 0 = Custom Sponge
6 = Sponge, 0 = Sponge Add
6 = Sponge, 0 = Sponge Remove
7 = Tint, 0 = Custom Tint
7 = Tint, 0 = Tint Lightly
7 = Tint, 0 = Tint A Lot
8 = Blend, 0 = Custom Blend
8 = Blend, 0 = Blend A Little
8 = Blend, 0 = Blend A Lot
9 = Sharpen, 0 = Custom Sharpen
9 = Sharpen, 0 = Sharpen A Little
9 = Sharpen, 0 = Sharpen A Lot

If ToolID = 27, BrushID and TypeID can be specified, respectively, as :
0 = Normal, 0 = Custom Normal
0 = Normal, 0 = Normal Clone
1 = Vango, 0 = Custom Impressionism
1 = Vango, 0 = Impressionism Clone
2 = Seurat, 0 = Custom Pointillism
2 = Seurat, 0 = Pointillism Clone
3 = From Saved, 0 = Custom From Saved
3 = From Saved, 0 = Light Eraser
3 = From Saved, 0 = Eraser
3 = From Saved, 0 = Scrambler

.TypeID Specifies the brush type, which varies depending on the tool type and the brush selected. See the BrushID settings above.

.MergeMode Specifies the Merge Mode:
0 = Normal
1 = Add
2 = Subtract
3 = Difference
4 = Multiply
5 = Divide
6 = Lighter
7 = Darker
8 = Texturize
9 = Color
10 = Hue
11 = Saturation
12 = Lum
13 = Invert
14 = And
15 = Or
16 = Xor
17 = Red
18 = Green
19 = Blue
20 = Cyan
21 = Magenta
22 = Yellow
23 = Black

.Amount Specifiies the rate of flow for tools like custom air brush and custom spray can.. Valid values range from 1 to 100.
Specifies the amount of flow for certain effect tools. Valid values range from -100 to +100 for some tools and 1 to 100 for others.

.NibShape Specifies the shape of the brush.
0 = Round
1 = Square
2 = Custom

.Size Specifies the size of the brush in pixels. Valid values range from 1 to 100.

.Transparency	Specifies the level of transmitted light that filters through the paint color. Valid values range from 0 to 99 percent.
.Rotate	Specifies degree of nib rotation. Valid values range from 0 to 360.
.Flatten	Specifies the flattening effect on the current nib. Valid values range from 1 to 100.
.SoftEdge	Specifies the amount of softness to apply to the edge. Valid values range from 0 to 100 percent.

Example

```
.BrushSettings 25, 0, 0, 0, 20, 0, 35, 0, 0, 100, 10
```

The above example sets the tool type to paint with a flow rate of 20, a brush size of 35 pixels, with a flat edge and 10 percent softness applied to the edge.

{button ,AL(`PP_Tools_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.FileSelectPartialArea (PHOTO-PAINT)

This command defines a specific area of the image to open.

Syntax

.FileSelectPartialArea .Left=*long*, .Top=*long*, .Right=*long*, .Bottom=*long*

Syntax	Definition
.Left	Specifies the X coordinate of the upper-left corner of the file's bounding rectangle in pixels, relative to the origin.
.Top	Specifies the Y coordinate of the upper-left corner of the file's bounding rectangle in pixels, relative to the origin.
.Right	Specifies the X coordinate of the lower-right corner of the file's bounding rectangle in pixels, relative to the origin.
.Bottom	Specifies the Y coordinate of the lower-right corner of the file's bounding rectangle in pixels, relative to the origin.

Note

File must have been created or opened as a partial file to use this command. The specified rectangle must be within the image.

Example

```
.FileSelectPartialArea 120, 168, 335, 239
```

The above example opens the specified area of the image.

{button ,AL(`PP_File_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.EffectPlugin (PHOTO-PAINT)

This command activates the specified plug-in effect.

Syntax

.EffectPlugin .GroupName=*string*, .EffectName=*string*, .MemoryType=*long*, .MemorySize=*long*, .Parameters=*string*

Syntax	Definition
.GroupName	Specifies the name of the effect group. Refer to the Effects menu for a complete list of effect groups.
.EffectName	Specifies the name of the effect. Refer to the Effects menu drop-down menu for a list of available effect names.
.MemoryType	Specifies the memory type. 0 = None 1 = Handle 2 = Pointer
.MemorySize	Specifies the size of memory in bytes.
.Parameters	Specifies the string of hexadecimal characters.

Note

The group and effect names may be different in different languages which may result in non-portability of scripts using this command.

Example

```
.EffectPlugin "3D Effects", "Emboss...", 1, 56,
"db00000001000000030000000000000001000000010000003200000032000000 7f909905000000000000000000000000
0000000000000000"
```

The above example activates the 3-D effect, Emboss.

{button ,AL(`PP_Tools_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.FileAcquireWithFile (PHOTO-PAINT)

This command lets you open a scanned image from a file and apply color correction to it. The color correction applied depends on the scanner originally used to scan the image.

Syntax

.FileAcquireWithFile .FileName=*string*, .Left=*long*, .Top=*long*, .Right=*long*, .Bottom=*long*, .LoadType =*long*

Syntax	Definition
.FileName	Specifies the name of the file.
.Left	Specifies the X coordinate of the upper-left corner of the file's bounding rectangle in pixels, relative to the origin.
.Top	Specifies the Y coordinate of the upper-left corner of the file's bounding rectangle in pixels, relative to the origin.
.Right	Specifies the X coordinate of the lower-right corner of the file's bounding rectangle in pixels, relative to the origin.
.Bottom	Specifies the Y coordinate of the lower-right corner of the file's bounding rectangle in pixels, relative to the origin.
.LoadType	Indicates the image load type of the file: 0 = All (coordinates are not used) 1 = Partial 2 = Resample 3 = Crop

Example

```
.FileAcquireFromFile "C:\COREL60\CUSTOM\TEST.1.CPT", 0, 0, 0, 0, 0
```

This example opens the scanned image named "TEST.1.CPT".

{button ,AL(`PP_File_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.GetMaskRectangle (PHOTO-PAINT)

This function determines the coordinates of the active mask.

Syntax

.GetMaskRectangle .Left=*long*, .Top=*long*, .Right=*long*, .Bottom=*long*

Syntax	Definition
.Left	Specifies the X coordinate of the upper-left corner of the mask's bounding rectangle in pixels, relative to the origin.
.Top	Specifies the Y coordinate of the upper-left corner of the mask's bounding rectangle in pixels, relative to the origin.
.Right	Specifies the X coordinate of the lower-right corner of the mask's bounding rectangle in pixels, relative to the origin.
.Bottom	Specifies the Y coordinate of the lower-right corner of the mask's bounding rectangle in pixels, relative to the origin.

Example

```
.GetMaskRectangle l&, t&, r&, b&
MESSAGE l&
MESSAGE t&
MESSAGE r&
MESSAGE b&
```

The above example gets the coordinates of the mask's rectangle.

{button ,AL(`PP_Get_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.GetObjectIsEditable (PHOTO-PAINT)

This function returns a value indicating whether the specified object is editable.
TRUE (-1) = Editable
FALSE (0) = Not editable

Syntax
.GetObjectIsEditable .ObjectID=*long*

Syntax	Definition
.ObjectID	Identifies the object.

Example

```
status& = .GetObjectIsEditable(2)
MESSAGE status&
```

The above example determines whether the specified object is editable.

{button ,AL(`PP_Get_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.GetObjectIsSelected (PHOTO-PAINT)

This function returns a value indicating whether the specified object is selected:
TRUE (-1) = Selected
FALSE (0) = Not selected

Syntax
.GetObjectIsSelected .ObjectID=*long*

Syntax	Definition
.ObjectID	Identifies the object.

Example

```
status& = .GetObjectIsSelected(2)
MESSAGE status&
```

The above example determines whether the specified object is selected.

{button ,AL(`PP_Get_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.GetObjectIsVisible (PHOTO-PAINT)

This function returns a value indicating whether the specified object is visible or hidden:
TRUE (-1) = Visible
FALSE (0) = Hidden

Syntax
.GetObjectIsVisible .ObjectID=*long*

Syntax	Definition
.ObjectID	Identifies the object.

Example

```
status& = .GetObjectIsVisible(2)
MESSAGE status&
```

The above example determines whether the specified object is visible.

{button ,AL(`PP_Get_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.GetObjectRectangle (PHOTO-PAINT)

This function determines the coordinates of the bounding rectangle of the specified object.

Syntax

.GetObjectRectangle .ObjectID=*long*, .Left=*long*, .Top=*long*, .Right=*long*, .Bottom=*long*

Syntax	Definition
.ObjectID	Identifies the object.
.Left	Specifies the X coordinate of the upper-left corner of the object's bounding rectangle in pixels, relative to the origin.
.Top	Specifies the Y coordinate of the upper-left corner of the object's bounding rectangle in pixels, relative to the origin.
.Right	Specifies the X coordinate of the lower-right corner of the object's bounding rectangle in pixels, relative to the origin.
.Bottom	Specifies the Y coordinate of the lower-right corner of the object's bounding rectangle in pixels, relative to the origin.

Example

```
.GetObjectRectangle 2, l&, t&, r&, b&
MESSAGE l&
MESSAGE t&
MESSAGE r&
MESSAGE b&
```

The above example determines the coordinates of the specified object's rectangle.

{button ,AL(`PP_Get_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.GetSelectedObjectsRectangle (PHOTO-PAINT)

This function determines the coordinates of the bounding rectangle of the selected object(s).

Syntax

.GetSelectedObjectsRectangle .Left=*long*, .Top=*long*, .Right=*long*, .Bottom=*long*

Syntax	Definition
.Left	Specifies the X coordinate of the upper-left corner of the object's bounding rectangle in pixels, relative to the origin.
.Top	Specifies the Y coordinate of the upper-left corner of the object's bounding rectangle in pixels, relative to the origin.
.Right	Specifies the X coordinate of the lower-right corner of the object's bounding rectangle in pixels, relative to the origin.
.Bottom	Specifies the Y coordinate of the lower-right corner of the object's bounding rectangle in pixels, relative to the origin.

Example

```
.MaskRectangle 82, 146, 270, 248, 0, 0
.ObjectCreate 0
.GetSelectedObjectsRectangle l&, t&, r&, b&
MESSAGE l&
MESSAGE t&
MESSAGE r&
MESSAGE b&
```

The above example gets the coordinates of the selected object's bounding rectangle.

{button ,AL(`PP_Get_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.ObjectEditAll (PHOTO-PAINT)

This command makes all objects editable, including the background.

Syntax
.ObjectEditAll

Example
```
.ObjectEditAll
```

The above example makes all objects editable including the background.

{button ,AL(`PP_Object_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.ObjectEditSelected (PHOTO-PAINT)

This command allows you to perform editing commands on all selected objects.

Syntax
.ObjectEditSelected

Example
```
.ObjectEditSelected
```
The above example makes all selected objects editable.

{button ,AL(`PP_Object_Menu;;;;',0,"Defaultoverview",)} **Related Topics**

.ObjectEdit (PHOTO-PAINT)

This command makes the specified object editable or uneditable.

Syntax

.ObjectEdit .ObjectID=*long*, .Edit=*boolean*

Syntax	Definition
.ObjectID	Identifies the selected object.
.Edit	Set to TRUE(-1) to make the object editable. Set to FALSE (0) to make the object uneditable.

Note

Making an object editable also makes it visible if it is hidden.

Example

```
.ObjectEdit 3, -1
```

The above example makes the specified object (object3) editable.

{button ,AL(`PP_Object_Menu;;;;',0,"Defaultoverview",)} **Related Topics**

.FillSolid (PHOTO-PAINT)

This command applies the specified solid fill color.

Syntax

.FillSolid .ColorModel=*long*, .Color1=*long*, .Color2=*long*, .Color3=*long*, .Color4=*long*

Syntax	Definition
.ColorModel	Specifies the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Specifies the first color component for .ColorModel. For example, Red is the first color component for RGB. Click + for valid value ranges.

.Color2 Specifies the second color component for .ColorModel. For example, Green is the second color component for RGB. Click + for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.

.Color3	Specifies the third color component for .ColorModel. For example, Blue is the third color component for RGB. Click + for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.

.Color4 Specifies the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. Click + for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.

Example

```
.PenSettings 24, 10, 0, 0, 0, 0, 0
.ToleranceSettings 0, 10, 10, 10, 10
.FillSolid 5, 0, 0, 255, 0
.Fill 114, 69, 0
```

The above example applies a solid blue fill.

{button ,AL(`PP_Draw_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.FillFountainColor (PHOTO-PAINT)

This command sets the Fountain Fill Color.

Syntax

.FillFountainColor .ColorModel=*long*, .Color1=*long*, .Color2=*long*, .Color3=*long*, .Color4=*long*, .Position=*long*, .Index=*long*

Syntax	Definition
.ColorModel	Specifies the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Specifies the first color component for .ColorModel. For example, Red is the first color component for RGB. Click  for valid value ranges.

.Color2 Specifies the second color component for .ColorModel. For example, Green is the second color component for RGB. Click  for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.

.Color3	Specifies the third color component for .ColorModel. For example, Blue is the third color component for RGB. Click  for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.

.Color4 Specifies the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. Click  for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.

.Position	Specifies the position at which to set the color. If position is 0, then the start color is set. If position is 100, then the end color is set. For other values, a color at that position is added (or changed if one already exists at that position. Note: If position is not 0 or 100, Blend is forced to be custom.
.Index	Specifies the position of the color in the palette. Valid values range from 0 to 255.

Example

```
.PenSettings 24, 10, 0, 0, 0, 0, 0
.ToleranceSettings 0, 10, 10, 10, 10
.FillFountainColor 5, 0, 0, 255, 0, 0, 0
.FillFountainColor 5, 255, 0, 0, 0, 100, 1
.FillFountainApply 6, 2, 256, 180, 0, 50, 50, 50
.Fill 107, 45, 0
```

The above example sets the first fountain fill color to red and the second color to blue.

{button ,AL(`PP_Draw_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.FillFountainApply (PHOTO-PAINT)

This command applies the specified fountain fill.

Syntax

.FillFountainApply .Type=*long*, .Colors=*long*, .Steps=*long*, .Angle=*long*, .EdgePad=*long*, .Horizontal Offset=*long*, .VerticalOffset=*long*, .Midpoint=*long*

Syntax	Definition
.Type	Specifies the type of Fountain Fill to apply: 0 = Linear (default) 1 = Radial 2 = Conical 3 = Square
.Colors	Specifies the number of colors used for the fill. This number will correspond to the number of calls to the FillFountainColor command.
.Steps	Specifies the number of stripes you want. Lower values produce coarser fountains on screen which take less time to redraw. Valid values range from 2 to 256.
.Angle	Specifies the angle at which the fill is applied in tenths of degrees. Positive values will rotate the fill counter-clockwise, negative values will rotate it clockwise.
.EdgePad	Specifies the amount of padding to apply to the fill. Ignored for type 2. Valid values range from 0 to 45 percent.
.HorizontalOffset	Specifies the horizontal offset of the center of the fill. Valid values range from -100 to +100 percent. A value of -50% will place the center on the left edge of your object; a value of 50% will place it on the right edge.
.VerticalOffset	Specifies the vertical offset of the center of the fill. Valid values range from -100 to +100 percent. A value of -50% will place the center on the bottom edge of your object; a value of 50% will place it on the top edge.
.Midpoint	Specifies the midpoint between the two offsets.

Example

```
.PenSettings 24, 10, 0, 0, 0, 0, 0
.ToleranceSettings 0, 10, 10, 10, 10
.FillFountainColor 5, 255, 255, 0, 0, 0, 0
.FillFountainColor 5, 255, 0, 0, 0, 100, 1
.FillFountainApply 6, 2, 256, 180, 0, 50, 50, 50
.Fill 97, 43, 0
```

The above example applies a two-color fountain fill.

{button ,AL(`PP_Draw_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.FillTexture (PHOTO-PAINT)

This command applies the specified texture fill.

Syntax

.FillTexture .LibraryName=*string*, .TextureName=*string*, .StyleName=*string*

Syntax	Definition
.LibraryName	Specifies the name of the Texture Library.
.TextureName	Specifies the name of the texture.
.StyleName	Specifies the name of the style.

Example

```
.PenSettings 24, 10, 0, 0, 0, 0, 0
.ToleranceSettings 0, 10, 10, 10, 10
.FillTexture "Samples", "Pizzazz mineral", "Cosmic Minerals"
.Fill 61, 42, 0
```

The above example applies a texture from the "Samples" library named "Pizzazz mineral" in the "Cosmic Minerals" style.

{button ,AL(`PP_Draw_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.FillBitmap (PHOTO-PAINT)

This command applies the specified bitmap fill.

Syntax

.FillBitmap .BitmapName=*string*, .Width=*long*, .Height=*long*, .XOffset=*long*, .YOffset=*long*, .TileColumn=*long*, .TileOffset=*long*, .MaintainAspect=*boolean*, .Scale=*boolean*, .OriginalSize=*boolean*

Syntax	Definition
.BitmapName	Specifies the name of the bitmap file.
.Width	Specifies the width of the bitmap in pixels.
.Height	Specifies the height of the bitmap in pixels.
.XOffset	Specifies the amount of offset applied to the first tile along the x-axis. Valid values range from 0 to 100.
.YOffset	Specifies the amount of offset applied to the first tile along the y-axis. Valid values range from 0 to 100.
.TileColumn	Set to TRUE (-1) to enable column offset. Set to FALSE (0) to enable row offset.
.TileOffset	Specifies the amount of row or column offsets. Valid values range from 0 to 100.
.MaintainAspect	Set to TRUE (-1) to maintain the aspect. Set to FALSE (0) to alter the aspect.
.Scale	Set to TRUE (-1) to scale the bitmap. Set to FALSE (0) to not apply scaling.
.OriginalSize	Set to TRUE (-1) to maintain the original size. Set to FALSE (0) to alter the original size.

Example

```
.PenSettings 24, 10, 0, 0, 0, 0, 0
.ToleranceSettings 0, 10, 10, 10, 10
.FillBitmap "C:\COREL60\CUSTOM\TILES\TC1_002B.TIF", 254000, 254000, 0, 0, 0, 0, 0, 0, 0
.Fill 105, 35, 0
```

The above example applies the specified bitmap fill.

{button ,AL(`PP_Draw_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.MaskSettings (PHOTO-PAINT)

This command specifies the current mask tool and settings.

Syntax

.MaskSettings .ToolID=*long*, .DrawMode=*long*, .Feather=*long*, .AntiAlias=*boolean*

Syntax	Definition
.ToolID	Specifies the type of mask tool to activate: 1 = Rectangle mask 2 = Elliptical Mask 3 = Freehand mask 4 = Mask noose 5 = Mask wand 8 = Mask pick
.DrawMode	Specifies the draw mode. 0 = Normal 1 = Add 2 = Subtract 3 = XOR
.Feather	Specifies the feathering width in pixels. Valid values range from 0 to 100.
.AntiAlias	Set to TRUE (-1) to apply anti-aliasing. Set to FALSE (0) to disable anti-aliasing.

Example

```
.MaskSettings 3, 0, 0, 0
```

The above example sets the freehand mask tool.

{button ,AL(`PP_Mask_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.ImageColorTable (PHOTO-PAINT)

This command ends a sequence of .PaletteColor commands used to create a palette. It also displays each color in a 16 or 256-color image. You can change an individual color or blocks of color in the table. When you change colors in the table, any incidences of those colors in the image are also changed.

Syntax
.ImageColorTable

Note

The .ImageColorTable command must be preceded by a .StartPalette command and a contiguous block of .PaletteColor commands.

Example

```
.StartPalette 16
.PaletteColor 5, 0, 0, 0, 0, 0
.PaletteColor 5, 128, 0, 0, 0, 1
...
...
...
.PaletteColor 5, 0, 255, 255, 0, 14
.PaletteColor 5, 255, 255, 255, 0, 15
.ImageColorTable
```

The above example creates a 16-color custom palette.

{button ,AL(`PP_Image_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.PaletteColor (PHOTO-PAINT)

This command sets the colors in a palette.

Syntax

.PaletteColor .ColorModel=*long*, .Color1=*long*, .Color2=*long*, .Color3=*long*, .Color4=*long*, .Index=*long*

Syntax	Definition
.ColorModel	Specifies the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Specifies the first color component for .ColorModel. For example, Red is the first color component for RGB. Click  for valid value ranges.

.Color2 Specifies the second color component for .ColorModel. For example, Green is the second color component for RGB. Click  for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0

.Color3	Specifies the third color component for .ColorModel. For example, Blue is the third color component for RGB. Click  for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.

.Color4 Specifies the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. Click  for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.

.Index	Specifies the position of the color in the palette.

Note

The .PaletteColor command must be in a contiguous block of .PaletteColor commands. The first .PaletteColor command in the block must be preceded by the .StartPalette command, and the last must be followed by the .ImageColorTable command. The .Colors value specified in the .StartPalette command sets the number of .PaletteColor commands that must follow the .StartPalette command.

Example

```
.StartPalette 16
.PaletteColor 5, 0, 0, 0, 0, 0
.PaletteColor 5, 128, 0, 0, 0, 1
...
...
...
.PaletteColor 5, 0, 255, 255, 0, 14
.PaletteColor 5, 255, 255, 255, 0, 15
.ImageColorTable
```

The above example creates a 16-color custom palette.

{button ,AL(`PP_Image_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.ContinueBezier (PHOTO-PAINT)

This command continues the Bezier drawing mode.

Syntax

.ContinueBezier .ptX=*long*, .ptY=*long*, .Closed=*boolean*, .Continuity=*long*, .Type=*long*

Syntax	Definition
.ptX	Specifies the X coordinate of the next Bezier point in pixels, relative to the origin.
.ptY	Specifies the Y coordinate of the next Bezier point in pixels, relative to the origin.
.Closed	Set to TRUE (-1) to draw a closed curve. Set to FALSE (0) to draw an open curve.
.Continuity	Specifies how the curve is derived. 0 = First order integral 1 = Second order integral
.Type	Specifies the node type. 0 = Line 1 = Curve 2 = Normal 3 = Control point

Note

The .ContinueBezier command must be in a contiguous block of one or more .ContinueBezier commands. The first .ContinueBezier command in the block must be preceded by the .StartBezier command, and the last must be followed by the .EndBezier command.

Example

```
.StartBezier 19
.ContinueBezier 96, 82, 0, 0, 3
.ContinueBezier 102, 85, 0, 1, 2
...
...
...
.ContinueBezier 235, 215, 0, 0, 3
.ContinueBezier 236, 221, 0, 0, 2
.EndBezier
```

The above example creates a 19-point Bezier curve.

{button ,AL(`PP_Draw_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.EndBezier (PHOTO-PAINT)

This command ends the Bezier drawing mode.

Syntax
.EndBezier

Note

The EndBezier command must be preceded by a .StartBezier command and a contiguous block of one or more .ContinueBezier commands.

Example

```
.StartBezier 19
.ContinueBezier 96, 82, 0, 0, 3
.ContinueBezier 102, 85, 0, 1, 2
...
...
...
.ContinueBezier 235, 215, 0, 0, 3
.ContinueBezier 236, 221, 0, 0, 2
.EndBezier
```

The above example creates a 19-point Bezier curve.

{button ,AL(`PP_Draw_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

.StartBezier (PHOTO-PAINT)

This command starts the Bezier drawing mode.

Syntax
.StartBezier .Points=*long*

Syntax	Definition
.Points	Specifies the number of Bezier points.

Note

The .StartBezier command must be followed by one or more .ContinueBezier commands, and one .EndBezier command.

Example

```
.StartBezier 19
.ContinueBezier 96, 82, 0, 0, 3
.ContinueBezier 102, 85, 0, 1, 2
...
...
...
.ContinueBezier 235, 215, 0, 0, 3
.ContinueBezier 236, 221, 0, 0, 2
.EndBezier
```

The above example creates a 19-point Bezier curve.

{button ,AL(`PP_Draw_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.StartPalette (PHOTO-PAINT)

This command initializes a series of custom palette creation statements. It must be used with the .PaletteColor and .ImageColorTable commands.

Syntax

.StartPalette .Colors=*long*

Syntax	Definition
.Colors	Specifies the number of colors used in the palette. Valid values range from 1 to 256. The value specified in .Colors sets the number of .PaletteColor commands that must follow the .StartPalette command.

Note

The .StartPalette command must be followed by a contiguous block of .PaletteColor commands and one .ImageColorTable command.

Example

```
.StartPalette 16
.PaletteColor 5, 0, 0, 0, 0, 0
.PaletteColor 5, 128, 0, 0, 0, 1
...
...
...
.PaletteColor 5, 0, 255, 255, 0, 14
.PaletteColor 5, 255, 255, 255, 0, 15
.ImageColorTable
```

The above example creates a 16-color custom palette.

{button ,AL(`PP_Image_Menu;;;;;',0,"Defaultoverview",)} Related Topics

.BindToActiveDocument (PHOTO-PAINT)

This command creates a link between the executing script and the active PHOTO-PAINT document. This command is used in cases when the user manually changes the active document during script execution.

Syntax
.BindToActiveDocument

Example

```
.FileNew 300, 300, 1, 100, 100, 0, 0, 0, -1, -1, -1, -1, 255, 0, 0, 0
.FileNew 300, 300, 1, 100, 100, 0, 0, 0, -1, -1, -1, -1, 0, 255, 255, 0
.FileNew 300, 300, 1, 100, 100, 0, 0, 0, -1, -1, -1, -1, 255, 0, 255, 0
MESSAGE "Select the image where you want to draw an ellipse"
.BindToActiveDocument
.SetPaintColor 5, 0, 0, 0, 0
.PenSettings 21, 3, 0, 0, 0, 0, 0
.ShapeSettings 0, 0, 3, -1, 0
.FillSolid 5, 255, 255, 0, 0
.Ellipse 200, 200, 250, 250
```

The above example creates three new documents, prompts the user to select one of the three, then creates a link to the selected document.

{button ,AL(`PP_File_Menu;;;;;',0,"Defaultoverview",)} **Related Topics**

ID	Color Model	Color 1	Color 2	Color 3	Color 4
1	Pantone	Pantone ID number	Tint (0 - 100)	Ignored	Ignored
2	CMYK100	Cyan (0 - 100)	Magenta (0 - 100)	Yellow (0 - 100)	Black (0 - 100)
3	CMYK255	Cyan (0 - 255)	Magenta (0 - 255)	Yellow (0 - 255)	Black (0 - 255)
4	CMY	Cyan (0 - 255)	Magenta (0 - 255)	Yellow (0 - 255)	Ignored
5	RGB	Red (0 - 255)	Green (0 - 255)	Blue (0 - 255)	Ignored
6	HSB	Hue (0 - 360)	Saturation (0 - 255)	Brightness (0 - 255)	Ignored
7	HLS	Hue (0 - 360)	Lightness (0 - 255)	Saturation (0 - 255)	Ignored
8	Black and White	Black (0) or White (1)	Ignored	Ignored	Ignored
9	Grayscale	Black % (0-255)	Ignored	Ignored	Ignored
10	YIQ255	Y-luminance (0 - 255)	I-chromaticity (0 - 255)	Q-chromaticity (0 - 255)	Ignored
11	L*a*b*	L*-lightness (0 - 255)	a*-green to red (0 - 255)	b*-blue to yellow (0 - 255)	Ignored

ID	Color Model	Color 1	Color 2	Color 3	Color 4
3	CMYK255	Cyan (0 - 255)	Magenta (0 - 255)	Yellow (0 - 255)	Black (0 - 255)
5	RGB	Red (0 - 255)	Green (0 - 255)	Blue (0 - 255)	Ignored
8	Black and White	Black (0) or White (1)	Ignored	Ignored	Ignored
9	Grayscale	Black % (0-255)	Ignored	Ignored	Ignored